ELSEVIER

Artificial Intelligence 119 (2000) 191–233

Artificial Intelligence

www.elsevier.com/locate/artint

The Spatial Semantic Hierarchy ☆

Benjamin Kuipers [1]

Computer Science Department, University of Texas at Austin, Austin, TX 78712, USA

Received 5 April 1999; received in revised form 5 December 1999

Abstract

The Spatial Semantic Hierarchy is a model of knowledge of large-scale space consisting of multiple interacting representations, both qualitative and quantitative. The SSH is inspired by the properties of the human cognitive map, and is intended to serve both as a model of the human cognitive map and as a method for robot exploration and map-building. The multiple levels of the SSH express states of partial knowledge, and thus enable the human or robotic agent to deal robustly with uncertainty during both learning and problem-solving.

The control level represents useful patterns of sensorimotor interaction with the world in the form of trajectory-following and hill-climbing control laws leading to locally distinctive states. Local geometric maps in local frames of reference can be constructed at the control level to serve as observers for control laws in particular neighborhoods. The causal level abstracts continuous behavior among distinctive states into a discrete model consisting of states linked by actions. The topological level introduces the external ontology of places, paths and regions by abduction to explain the observed pattern of states and actions at the causal level. Quantitative knowledge at the control, causal and topological levels supports a "patchwork map" of local geometric frames of reference linked by causal and topological connections. The patchwork map can be merged into a single global frame of reference at the metrical level when sufficient information and computational resources are available.

We describe the assumptions and guarantees behind the generality of the SSH across environments and sensorimotor systems. Evidence is presented from several partial implementations of the SSH on simulated and physical robots. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Spatial reasoning; Cognitive map; Robot exploration; Map learning; Qualitative reasoning

☆ This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-9504138 and CDA 9617327, by NASA grant NAG 9-898, and by the Texas Advanced Research Program under grants 003658-242 and 003658-347. Thanks to my former students Yung-Tai Byun, Rick Froom, Bert Kay, Wan-Yik Lee, and David Pierce; to my current students Rob Browning, Bill Gribble, Mike Hewett, and Emilio Remolina; and to two anonymous reviewers.

[1] Email: kuipers@cs.utexas.edu.

PII: S0004-3702(00)00017-5

1. Introduction

1.1. Why spatial knowledge?

Spatial knowledge is foundational to commonsense knowledge, and hence to most kinds of knowledge that humans possess. Spatial metaphors are ubiquitous in discourse, and draw on preexisting spatial knowledge to communicate relationships and processes that would be difficult to communicate otherwise [59]. Spatial knowledge is grounded in sensorimotor experience: that is, the meanings of symbols in a symbolic representation of space are constrained by experience of perception and action [33]. Spatial knowledge takes a number of quite different forms, including procedures for getting from one place to another, topological network maps of an environment, and geometrical models of the environment [67]. Children exhibit qualitatively different types of behavior as they grow and develop, acquiring the ability to construct and use different forms of spatial knowledge [79,89].

There are several different types of spatial knowledge, distinguished by the nature of the interaction between the agent and the environment. This paper focuses primarily on *large-scale space*, which is defined as space whose structure is at a much larger scale than the sensory horizon of the agent. Thus, to learn a map, the agent must travel through the space, gathering local observations and inferring their global relationships from the actions linking them. Memory and processing limitations are also important to the representation of large-scale space, since the time required for travel is long enough for other pressing concerns to interrupt the processing of spatial knowledge. Human knowledge of large-scale space is sometimes called the *cognitive map*, though it is in many ways not map-like [48]. Other closely related but distinct types of spatial knowledge include *visual space*, which describes the immediately surrounding environment and is explored quickly by moving the gaze [97], and *graphical space*, a special case of visual space where the structure is the spatial layout and relations among symbols on paper or other display [30].

This paper focuses on large-scale space because the different types of knowledge and the learning processes are particularly accessible, being spread out in space and time, both exploration time and developmental time. We also focus on the problems of exploring the environment and learning its spatial structure, rather than on problem-solving, on the grounds that many effective algorithms exist for way-finding based on various spatial representations [20,35].

1.2. The Spatial Semantic Hierarchy

We propose that knowledge of large-scale space consists of several distinct but interacting representations, each with its own ontology, collectively known as the Spatial Semantic Hierarchy (SSH). The multiplicity of representation in the SSH gives the agent more expressive power for incomplete knowledge, and more robustness in coping with sensorimotor uncertainty and computational limitations, compared with any individual representation. The goals of this paper are

(1) to describe the multiple representations of the Spatial Semantic Hierarchy, and their rationale, with sufficient clarity that they can be implemented by the qualified reader; and
(2) to demonstrate that the multiple representations can work together coherently and effectively, in part by exhibiting several different partial implementations of the SSH.

The SSH representations can be arranged in a lattice (Fig. 1). Each node corresponds to a representation, which is specified in terms of its ontology (the set of objects and relations that can be represented), and a set of axioms and inference rules that determine what conclusions can be inferred and what actions can be taken. Closed-headed arrows represent dependencies, meaning that knowledge in the representation at the head of the arrow presupposes, or is defined in terms of, or is inferred from, knowledge in the representation at the tail of the link. It is these dependencies that justify calling the SSH a *hierarchy* of representations. We predict that individual variation (with developmental stage, exploration experience, or cognitive style) must necessarily respect the dependencies in this lattice, but testing this hypothesis is beyond the scope of this paper. Open-headed arrows represent paths of potential information flow, but not dependencies.

The nodes of the lattice are also structured in two independent dimensions. First, much spatial knowledge is qualitative rather than quantitative, meaning that continuous quantities are represented by descriptions that can be manipulated as nominal or ordinal quantities, rather than as interval or ratio quantities [92].[2] Furthermore, quantitative knowledge can be subdivided into continuous-valued attributes that can be represented and manipulated within a symbolic theory, and high-resolution analog models that mimic properties of the space itself.

The second dimension (vertical in Fig. 1) organizes the spatial knowledge representations into levels according to ontology. The inclusion of both qualitative and quantitative knowledge at most levels of the SSH is a significant change from previous presentations [52,56,58].

The sensory and control levels deal with continuous sensing of a continuous world, and produce continuous behavior. The transition to the causal level abstracts continuous behavior to discrete states and actions. The transition to the topological level does an abduction, hypothesizing places and paths to account for observed states and actions. A global metrical map is created by merging local geometric maps as linked by the topological description. Quantitative information is useful at every level when it is available, but effective behavior is often possible with only qualitative knowledge.

The *sensory level* is the interface to the agent's sensory system. Our primary focus is on motion and exploration guided by continuous sensors such as vision, laser or sonar range-sensing. However, qualitative sensory input in the form of designating names could be added very naturally. Structured communication through maps or verbal commands are discussed separately (Section 8.3).

[2] A quantity can be represented by several different abstract datatypes, distinguished by which operations can be applied. A nominal quantity supports only match for equality. Ordinal quantities support comparison for greater-than, less-than, or equal-to. Interval quantities also support the difference operation, and zero is only an arbitrary landmark. Ratio quantities have a true zero value and support multiplication and division by scalars. Qualitative reasoning systems such as QSIM [51] reason primarily with ordinal abstractions of quantities.

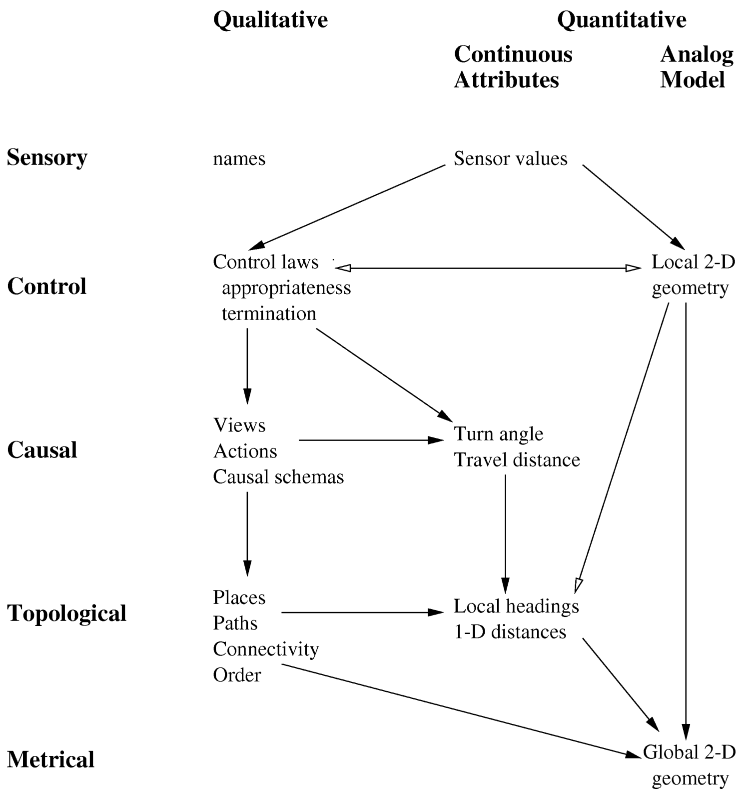


Fig. 1. The distinct representations of the Spatial Semantic Hierarchy. Closed-headed arrows represent dependencies; open-headed arrows represent potential information flow without dependency.

The *control level* describes the world in terms of continuous control laws that bind the agent and its environment into a dynamical system throughout a qualitatively uniform segment of the environment. Associated with each control law are conditions for its appropriateness, and for its termination once it has been selected. Local geometric maps can also be created at the control level, to function as observers of the local environment for the control laws, more powerful than individual sensory features.

An agent can be unambiguously localized within a local neighborhood by selecting a hill-climbing control law that moves it physically to an isolated distinctive state. A trajectory-following control law takes the agent from one distinctive state to a neighborhood where hill-climbing can bring it to the next distinctive state. If a local geometric model of the neighborhood exists, it may be possible for the agent to localize itself within the model without physically moving to a distinctive state. Section 2 formalizes the sensory and control levels using the continuous mathematics of dynamical systems.

The *causal level* abstracts the continuous world, and the agent's behavior within it, to a discrete model described in terms of sensory views, actions, and the causal relations among them. This ontological change abstracts away the details of how views are defined or how actions are implemented in particular circumstances. The magnitudes of turn and travel actions may be described by simple quantitative attributes. Plans made at the causal level are straight-forwardly translated down to the control level for execution and monitoring. Section 3 describes the causal level in more detail.

The *topological level* introduces the ontology of places, paths and regions, and their connectivity, order and containment relations: features of an external environment. The topological model of the environment is constructed by the non-monotonic process of *abduction*, positing the minimal set of places and paths needed to explain the regularities observed among views and actions at the causal level. A topological network map, particularly one augmented with a hierarchical region structure, is much more effective for planning than the flat causal action model. The topological map can be augmented with quantitative attributes to improve planning further, but the ability to plan and act is not dependent on the availability of quantitative spatial knowledge. Section 4 describes the topological level in more formal detail in terms of first-order logic.

The *metrical level* represents a global geometric map of the environment in a single frame of reference, which may be useful but is seldom essential. Quantitative spatial information is represented at each level of the hierarchy, from local analog maps at the control level, to action magnitudes at the causal level, to local headings and distances at the topological level. This is enough to represent a "patchwork metrical map" of local frames of reference linked by a topological network structure. Section 5 discusses the problem of unifying local frames of reference into a global metrical map, and when such a map is important.

Section 6 describes a number of implementations of portions of the SSH framework on both simulated and physical robots, that demonstrate how multiple representations can work effectively together, and which have motivated revisions to the framework. Section 7 discusses practical issues of matching the general SSH framework to the sensors and effectors of a particular robot, and Section 8 discusses a variety of related questions.

1.3. Our previous work

The challenge to designing a hierarchical model like the SSH is finding the natural joints to dissect the complex natural phenomena of spatial knowledge. This paper re-presents, reorganizes, revises and extends our previous work [46,47,52,55–58,62,81]. Preliminary versions of the formalization in Sections 2–4 appeared in [52], which was reprinted in [53]. These have been revised and extended, and the treatment of metrical knowledge has been dramatically changed.

Kuipers and Byun [55,56] implemented the SSH control, topological, and metrical levels [3] on a simulated robot with a radial array of 16 range-sensors subject to both random and systematic errors similar to those of the Polaroid sonar sensor. Fig. 2(a)

[3] The causal level was incorporated later [52], drawing on much earlier work in the TOUR model [47], driven by the recognition that control laws converge to distinctive *states* (x, y, θ) in the configuration space of the robot, rather than to distinctive *places* (x, y) in the environment. Distinctive states may be linked by turn actions, which therefore correspond to trajectory-following control laws in configuration space.

shows the trajectories followed and the distinctive places identified as the robot explored its environment and created the topological map shown (in part) in Fig. 2(b), with edges and places annotated with the names of their respective trajectory-following and hill-climbing control laws. Careful examination of the trajectories in Fig. 2(a) reveals the exploration strategies used to disambiguate systematic sensory errors (i.e., specular reflections) and locally indistinguishable places. Local metrical information is also accumulated in the form of local maps of place neighborhoods and generalized cylinder models of edges. When this patchwork metrical map is relaxed into a single global frame of reference, the result shown in Fig. 2(c) is a good match for the original environment in Fig. 2(a).

Since the SSH is intended to describe knowledge of large-scale space in both humans and robots, this paper refers to the "agent" or "traveller" unless referring specifically to a "robot" as engineered artifact.

2. The control level

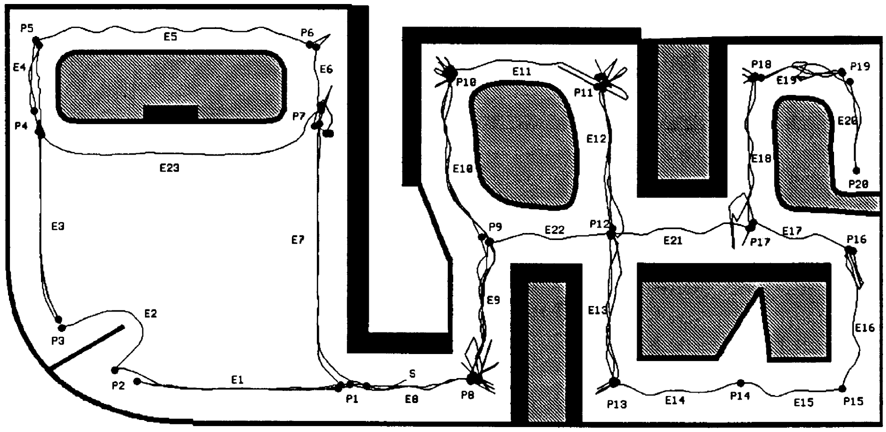
The SSH control level builds a map of the environment by identifying a set of overlapping, qualitatively uniform segments of the state space of the agent in the environment. Each segment is qualitatively uniform in the sense that the control law (e.g., "Follow the right wall" or "Approach visual target") associated with that segment will bring the agent close to a particular final state, typically within the overlap between two or more segments.

For most of this section, we assume that the agent can only use sensor input to drive control laws, and cannot build or use metrical maps. (We will return to local metrical maps in Section 2.5.) The ability of the control level to support a useful cognitive map even under such a restrictive assumption contributes significantly to its robustness.

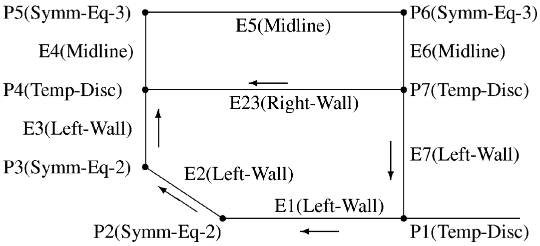
At the control level, the agent receives a continuous stream of time-varying sense values and outputs a continuous stream of motor signals. A control law specifies the relation between sensory input and motor output. The agent, its environment, and the currently selected control law form a continuous dynamical system, which can be modeled by a differential equation, and whose behavior is described by the solution to that differential equation.

Exploration of an unknown environment takes place by selecting a control law based on sensory information available about the local neighborhood. Typically, we expect behavior to be an alternation between *hill-climbing* control laws, which bring the agent to a locally-distinctive state from any state within the local neighborhood, and *trajectory-following* control laws, which bring the agent from one distinctive state to the neighborhood of the next (Fig. 3).

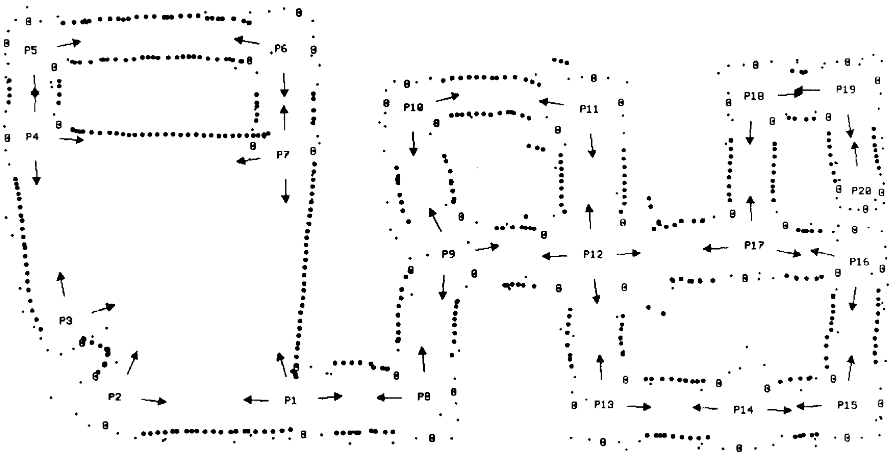
A *locally distinctive state* within a neighborhood is a uniquely determined state that the agent converges to by following a single control law. Typically, these are hill-climbing control laws seeking the isolated local maximum of a *distinctiveness measure*. For example, most of the distinctive states in Fig. 2(a) are determined by a d-measure whose local maximum is equidistant from nearby obstacles. However, some distinctive states are determined by the points along a trajectory where a sudden change takes place, such as P1, P4, and P7 in Fig. 2(a). The set of isolated distinctive states, connected by trajectory-



(a)



(b)



(c)

Fig. 2. The simulated NX robot applies the SSH mapping strategy. (a) The exploration trace shows distinctive places and paths. (b) The topological map (fragment) identifies places and paths. (c) The global metrical map. Reprinted from Robotics and Autonomous Systems, Vol. 8, B.J. Kuipers and Y.-T. Byun, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations, pp. 59–60 (1991), with permission from Elsevier Science.

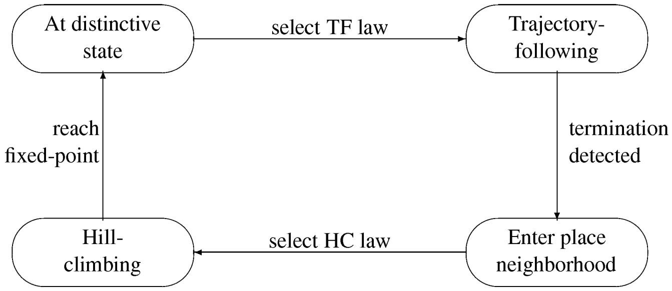


Fig. 3. Distinctive places found by alternating trajectory-following and hill-climbing control laws.

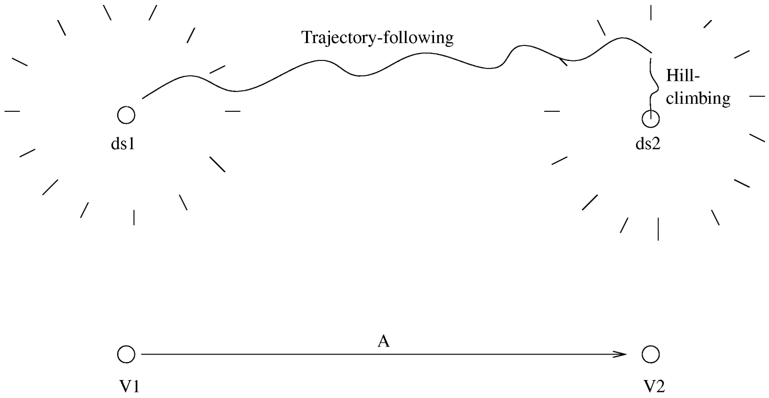


Fig. 4. Abstraction from controlled behavior to causal link $\langle V_1, A, V_2\rangle$.

following then hill-climbing, is the key to the abstraction from continuous behavior in a continuous world to a symbolic map of places and paths.

Cumulative error is the bane of robot map-making. Sensor and motor errors are inevitable. However, for travel between distinctive states, the error that accumulates during trajectory-following is reduced by hill-climbing below any desired tolerance (Fig. 4). Once travel from one distinctive state to another is reliable, i.e., accuracy is consistently good enough to reach the neighborhood of the destination state, the behavior pattern can be abstracted to the causal level.

The SSH control level requires the agent's representation to include a set of control laws, a selection method for determining the most appropriate control law for the current segment, and conditions for each control law that specify when the agent is approaching the end of the qualitatively uniform segment. For example, if the agent is at a distinctive state facing a direction without obstruction, but with walls on both sides, the most appropriate control law may be "Follow-the-Midline", which continues to apply until an obstruction appears to block travel or one or both side walls disappears.

After the cognitive map of a new environment has been learned, a route between two places can be found in many different ways, for example by metrically-guided graph search in the topological graph of places and paths. The route is translated down to the control level where it can determine the agent's behavior, by specifying a sequence of control laws in overlapping qualitatively uniform segments of state space.

2.1. Viewing the agent as a dynamical system

The agent has an objective location in the environment, but it does not have direct access to a representation of that location in an absolute frame of reference. Assume that the environment is two-dimensional, so that the *state* of the agent has three dimensions: position (x, y) and orientation θ. The vector of state variables is $\boldsymbol{x} = [x, y, \theta]$. The agent also has a memory M including symbolic descriptions of goals, beliefs, etc., which can influence the choice of control law, hence behavior.

The agent has a vector of sensors providing input $\boldsymbol{s} = [s_0, \ldots, s_{n-1}]$ and a vector of motor outputs $\boldsymbol{u} = [u_0, \ldots, u_{k-1}]$ by which it can change its position in the environment.

The sensor values are a function of the agent's state,

$$[s_0, \ldots, s_{n-1}] = \boldsymbol{s} = \Psi(\boldsymbol{x}) = \Psi(x, y, \theta). \tag{1}$$

All variables are piece-wise continuous functions of time. This model treats the environment as static, with the only changes being to the agent's position and orientation.

The "physics of the environment" (or dynamics of the agent),

$$[\dot{x}, \dot{y}, \dot{\theta}] = \dot{\boldsymbol{x}} = \Phi(\boldsymbol{x}, \boldsymbol{u}) = \Phi(x, y, \theta, u_0, \ldots, u_{k-1}) \tag{2}$$

specifies how the state, and hence the sensory values, change with time as a function of the current state and the motor outputs. The agent does not have direct access to its state vector $\boldsymbol{x}$, but only to the sensory information $\boldsymbol{s}(t)$ provided to it as it moves through the environment.

During a particular segment i of reactive behavior, the agent moves through the environment by setting its motor vector in response to its sensory inputs, according to a control law χ_i.

$$[u_0, \ldots, u_{k-1}] = \boldsymbol{u} = \chi_i(\boldsymbol{s}) = \chi_i(s_0, \ldots, s_{n-1}). \tag{3}$$

A control law χ_i is reactive in that it takes $\boldsymbol{s}(t)$ as input and produces $\boldsymbol{u}(t)$ as output, but it may also maintain a certain amount of local state information. For example, to implement a PID (proportional-integral-derivative) control law, χ_i must determine e, $\int e$, and $\dot{e}$, where $e(t)$ is an error term representing deviation from a setpoint value, computed from information in $\boldsymbol{s}(t)$.

For a given choice of control law χ_i, Eqs. (1), (2) and (3) define a dynamical system that describes the behavior of the agent interacting with its environment (Fig. 5).

2.2. Selecting the control law

Different strategies have been used to select control laws in different implementations of the SSH. Kuipers and Byun [55,56] used a simple rule-based system, and Lee [62] used a decision-tree, based on perceived features of the local environment. Kuipers [52] proposed to combine all relevant control laws, weighted by their degree of appropriateness, in the spirit of fuzzy control or heterogeneous control [54]. Pierce [80,81] learned *local state*

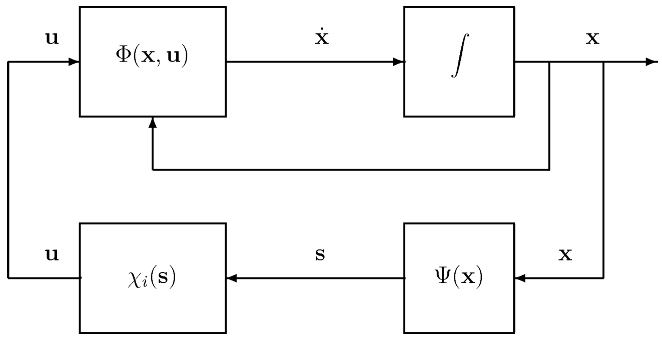


Fig. 5. Within any given qualitatively uniform segment of the environment, the robot behaves according to a block diagram representing Eqs. (1), (2) and (3) in Section 2.1. The selection of the control law χ_i is the external input that determines the robot's behavior.

variables—sensory features defined in the local environment that could be used as state variables—and selected control laws based on the number and independence of local state variables.

Other compositional approaches to control include potential field methods [2,90] and fuzzy control [41,68]. Appropriateness measures and other parameters of the control laws χ_i may be acquired and optimized by function-learning methods including neural nets (e.g., [82]) and memory-based learning [3,4].

2.3. Putting control into action

While the agent cannot sense its state vector $\boldsymbol{x}$ directly, within each qualitatively uniform segment it can define a vector $\boldsymbol{y}$ of *local state variables* which is at least partially determined by information in the sense vector [81]. For example, when a sonar-sensing robot is following a wall, its lateral position and orientation are directly observable, while its longitudinal position is not. We assume that there are locally-meaningful versions of Eqs. (1) and (2):

$$\boldsymbol{s} = \Psi(\boldsymbol{y}), \tag{4}$$

$$\dot{\boldsymbol{y}} = \Phi(\boldsymbol{y}, \boldsymbol{u}). \tag{5}$$

When the control law $\boldsymbol{u} = \chi_i(\boldsymbol{s})$ is determined by a distinctiveness measure $d(\boldsymbol{s})$, it is natural to specify its intended effect as climbing the gradient of $d(\boldsymbol{s})$

$$\dot{\boldsymbol{y}} = \nabla d(\boldsymbol{s}) \tag{6}$$

with respect to the local state variables. The problem remains of translating $\dot{\boldsymbol{y}}$ into values for the agent's motor output variables $\boldsymbol{u}$.

In simple cases, the dynamics of the agent (Eq. (5)) will have a pseudo-inverse Φ^{-1} so that, given $\boldsymbol{y}$ and a desired $\dot{\boldsymbol{y}}$, we can directly compute

$$\boldsymbol{u} = \Phi^{-1}(\boldsymbol{y}, \dot{\boldsymbol{y}}) \text{ such that } \dot{\boldsymbol{y}} = \Phi(\boldsymbol{y}, \boldsymbol{u}). \tag{7}$$

In general (i.e., for a robot with non-holonomic motion constraints), there may be no way to achieve a desired $\dot{\boldsymbol{y}}$ for a given state $\boldsymbol{y}$ (cf. [60]). In such a case, we specify the

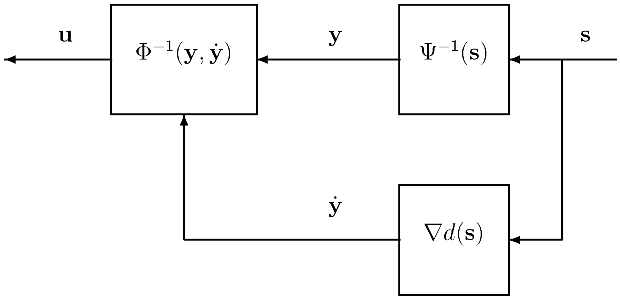


Fig. 6. One of many ways to realize the control law $\boldsymbol{u} = \chi_i(\boldsymbol{s})$ in Fig. 5, in case there is an observer $\Psi^{-1}(\boldsymbol{s})$ that describes the local environment in terms of a local state vector $\boldsymbol{y}$, the control law can be described as hill-climbing a distinctiveness measure $d(\boldsymbol{s})$, and there is a pseudo-inverse $\Phi^{-1}(\boldsymbol{y}, \dot{\boldsymbol{y}})$ that determines $\boldsymbol{u}$.

control goal as a net change $\Delta\boldsymbol{y}$ to be obtained over some period of time. Then we assume the ability to plan a sequence of continuous actions (e.g., [78]), or to retrieve a previously developed control plan:

$$p = plan(\boldsymbol{y}, \Delta\boldsymbol{y}), \text{ such that } \boldsymbol{u} = p(\boldsymbol{y}, t) \tag{8}$$

has the desired effect of reaching the state $\boldsymbol{y} + \Delta\boldsymbol{y}$. Note that, as with parallel parking, the intermediate states of the plan p may be farther from the goal than the initial or final states. Further extensions will be required to cope with pedestrians and other unexpected obstacles.

Eq. (6), along with either (7) or (8), provides an instance of the control law χ_i required by Eq. (3), as shown in Fig. 6. Thus, the agent's behavior *during a single hill-climbing or trajectory-following segment* consists of the state-evolution of a particular dynamical system. Higher-level symbolic reasoning intervenes at the joints between these segments to determine which dynamical system controls the behavior.

2.4. Smooth transitions between control laws

We have assumed that a single trajectory-following control law carries the robot from one distinctive state to the neighborhood of the next. However, travel from one distinctive state to another may require the combination of multiple control laws, with smooth transitions as each hands off to the next. For example, a robot starts facing "open space", first takes open-loop motion into the corridor it faces, then follows the midline to the end of the corridor. Upon reaching the end, the robot does hill-climbing to position itself equidistant from nearby obstacles.

We can accomplish these smooth transitions by assigning the control laws to overlapping operating regions characterized by fuzzy set membership functions, which we call "appropriateness measures" [54]. Where each control law $\chi_i(\boldsymbol{y})$ has an appropriateness measure $a_i(\boldsymbol{y})$, we take an appropriateness-weighted average.

$$\boldsymbol{u} = \frac{\sum_i a_i(\boldsymbol{y})\chi_i(\boldsymbol{y})}{\sum_i a_i(\boldsymbol{y})}. \tag{9}$$

When χ_i is not meaningful, $a_i(\boldsymbol{y}) = 0$. Note that as the agent moves, the effective number of participating local control laws may change.

2.5. Local 2-D geometry

Hill-climbing to distinctive states eliminates cumulative position error by bringing the agent to one of a discrete set of isolated states. When sufficient sensor input and computational resources are available, it becomes possible to build local geometric models of place neighborhoods and path segments. When the agent enters a place neighborhood, if it can localize itself with respect to a local metrical map of the neighborhood, physical hill-climbing to a distinctive state may be unnecessary. Localization provides its position in the frame of reference of the distinctive state, and orients it with respect to the trajectory by which it will depart from the neighborhood. It can therefore "cut the corner" and avoid the hill-climbing step while continuing along its route.

We say that an agent is *localized* within a place neighborhood, either because it has physically moved to a distinctive state in that neighborhood, or because it has identified its position and orientation with respect to a local metrical map of that neighborhood. These two cases play the same functional role: the agent can reliably initiate a trajectory-following control law to reach the neighborhood of the next distinctive state on its route.

Estimated position and accumulated error with respect to previous frames of reference may, if desired, simply be discarded. The local geometric maps form a patchwork of distinct frames of reference, linked by travel patterns that will later be described symbolically in the causal and topological maps.

There are many reliable methods for map-building and localization within local geometric models of place neighborhoods and path segments, including occupancy grids [73,94], sonar target maps [63], and generalized cylinder models [6,74]. By restricting their application to local neighborhoods, we avoid their vulnerability to cumulative estimated position error in large-scale and non-simply-connected spaces (cf. Fig. 7).

2.5.1. Occupancy grids

There is a pursuasive intuition that the "mind's eye" looks at a "map in the head" where spatial knowledge is represented and can be manipulated. Psychological studies such as Shepard and Metzler's classic mental rotation experiment [88] and Kosslyn's thorough exploration of mental imagery phenomena [42] provide convincing evidence of a cognitive capability in humans that can function as a spatial analog. Aspects of this cognitive functionality may come from neural mechanisms in the high level vision system [43]; some aspects may be attributed to neural mechanisms in the hippocampus, the seat of working memory [77]. (On the other hand, consideration of human states of partial knowledge makes it clear that the "map in the head" hypothesis cannot be adequate by itself [49].)

These observations led to proposals for computational models of spatial knowledge as grids of cells representing a fine-grained decomposition of the environment [44, 72,73]. In an *occupancy grid*, each cell holds a number representing the probability that the corresponding location in the environment is occupied. Regions of high values represent obstacles, low values represent free space, and intermediate values represent lack of knowledge: a representation well suited to the properties of sonar sensors [72]. In a recent formulation of occupancy grids, Thrun [94] alternates a *localization* phase, in which the robot estimates the likelihood of being in each location given its sensor readings and the existing map, followed by a *mapping* phase, in which the robot estimates the likelihood map of obstacles given its sensor readings and localization. Although this process converges quickly, the space and time costs of the representation and its computation are substantial. More fundamentally, although the localization phase can eliminate cumulative position error during local travel to known locations, it cannot prevent its accumulation during global exploration of new territory (e.g., the "around the block" scenario in Fig. 7).

The SSH addresses this problem by associating local 2-D spatial analog representations with individual place neighborhoods. In the topological map, these are linked by edges possibly annotated with their own metrical properties. Geometric localization applies only within individual place neighborhoods, where sensor information is available and relevant. Uncertainty about the global connectivity of the environment is detected and resolved when the topological map is constructed. This does not eliminate the global connectivity problem, but decouples it from the problem of cumulative position error. Breaking up the occupancy grid computation into small patches substantially reduces the computation time required, since occupancy grid algorithms are at best quadratic in the number of grid cells. Furthermore, the space requirement is also reduced because occupancy grids are needed only in the neighborhoods of topological places, not for the entire environment.

The occupancy grid is not the only possible representation for the local metrical map. An alternative, which we call the "sensory target map", is to identify, classify, and localize a small set of objects in the environment that have distinctive sensory properties [63,91]. Like the occupancy grid, the sensory target map defines location within a single frame of reference. Unlike the occupancy grid, whose size depends on the size of the environment and the desired precision of position descriptions, the size of the sonar target map depends on the number of targets represented and is independent of the precision required of positions. The localization-mapping cycle can be used in either representation.

Lee [62] found that the local metrical map could function as a reliable observer for control laws operating in a place neighborhood, compensating for a sparse and errorful sonar image. Certain critical features such as convex corners can be invisible to sonar sensors from many directions due to specular reflections off the walls. Lee's robot built a local sonar target map when moving into each new neighborhood. A feedback control law like *Move equidistant between wall and convex corner* could be executed using "virtual sensing" of the convex corner (i.e., dead reckoning using the local map) even when the corner could not be sensed directly. [4]

2.5.2. Generalized cylinders

While it is possible to represent the local metrical map of a path segment in the single 2-D frame of reference of an occupancy grid or sonar target map, the generalized cylinder representation [6,74] has the advantage that its different attributes are very loosely coupled, and are well-matched to the attributes of a path. For example, one can have good knowledge

[4] Often, with available sensors, the world is *not* its own best model.

of the length of the cylinder, and moderate knowledge of its cross-section, while having quite poor knowledge of the curvature of the axis.

Suppose the action of traversing a particular path segment from one distinctive state to another is described by (`travel` δ), where odometry gives us the measurement δ of the distance travelled. Over a number of such actions, our estimate of the true length L of the path segment will improve.

Given some degree of knowledge of L, we can define functions

$$l, r : [0, L] \to \mathbb{R}$$

so that $l(s)$ and $r(s)$ represent the minimum distance to an object on the left and right sides, respectively, when the agent is at point s along the trajectory. We can also define the function

$$c : [0, L] \to \mathbb{R},$$

where $c(s)$ represent the *curvature* of the trajectory travelled, at the point $s \in [0, L]$ along the path. Repeated observations of s, $l(s)$, $r(s)$, and $c(s)$ can be combined to estimate the most likely true values [27,91]. It is also possible to integrate knowledge of $s(t)$ and $c(s)$ to derive a trajectory $[x(t), y(t), \theta(t)]$ in state space.

The generalized cylinder representation supports a smooth progression from modeling a path segment purely as a topological link (defined by a control law like *Follow the midline*), to weak metrical information such as estimating the length of the link from odometry, to stronger metrical information such as estimating cross-section as a function of distance along the axis, to stronger yet by integrating estimates of the curvature of the axis. Eventually we may accumulate enough information to project into a single grid frame of reference (if desired), but each intermediate step along the way is useful.

2.6. Guarantees at the control level

We provide a guarantee for the purely qualitative control level, where localization is done by hill-climbing to distinctive states. The generalization to localization within local metrical maps is straight-forward.

The navigation strategy of alternating trajectory-following and hill-climbing control laws presumes that the following criteria are satisfied. We call these the *closure criteria* on the set of control laws.

(1) After a hill-climbing control law is executed and terminates at a distinctive state, at least one trajectory-following control law is available for selection. This ensures that there is a choice of action from the current distinctive state: there are no dead ends.

(2) After a trajectory-following control law is executed and reaches its termination state, at least one hill-climbing control law is available for selection. This ensures that each trajectory terminates at a distinctive state.

If the choice of hill-climbing control law is unique, or at least if every available choice brings the agent to the same distinctive state, then the abstraction from the control level to the causal level will be simple and deterministic. However, if closely-competing hill-climbing control laws bring the agent to different distinctive states, with different choices of departure trajectory, then the causal level will be non-deterministic.

When creating a particular robot for a particular environment, the robot's designer (or a learning algorithm) must ensure that the control laws χ_i satisfy the closure criteria, and preferably support a deterministic abstraction at the causal level.

2.7. Summary

There are several distinct benefits due to the SSH control level.

- By identifying and navigating among locally distinctive states, the control level makes the critical abstraction from continuous to discrete descriptions of behavior, necessary to support symbolic reasoning methods at higher levels.
- Distinctive states allow the agent to register its position with respect to the environment, and thus eliminate cumulative position error, without requiring a metrically accurate map or a global frame of reference.
- When local geometrical maps can be constructed, localization with respect to the frame of reference of a place neighborhood can substitute for hill-climbing to a distinctive state.
- The control level makes few assumptions about the agent's world or its sensorimotor system, and those it does make are generic mathematical properties of the agent-plus-environment considered as a dynamical system.
- The closure criteria can be used to evaluate particular sets of distinctiveness and appropriateness measures and control laws in particular environments. Even if they are violated occasionally, if the agent can detect violations, it may be able to recover, reorient, and continue on its way.

3. The causal level

When a sequence of control laws—trajectory-following then hill-climbing—reliably takes the agent from one distinctive state to another, we abstract the sequence of control laws to an action A, and the two distinctive states to the sensory images, or views, V and V', obtained there (Fig. 4). Their association is represented by the schema $\langle V, A, V'\rangle$.

We treat the case where the agent is localized with respect to a local metrical map as functionally equivalent to being at a distinctive state. In the view representation, the sensory image and its description is augmented with the agent's position and orientation within the local frame of reference.

When this abstraction can be applied throughout the environment, the continuous state space in which the agent is described as following the trajectories of a dynamical system is abstracted to a discrete state space in which the agent is described as performing a sequence of discrete actions resulting in state transitions. Situation calculus replaces differential equations as the most appropriate formalism.

3.1. Views, actions, and schemas

A *view* is a description of the sensory input vector $\boldsymbol{s}(t) = [s_1(t), \ldots, s_n(t)]$ obtained at a locally distinctive state. A view could be a complete snapshot of $\boldsymbol{s}(t)$, or it could be a partial description, consistent with more than one value of $\boldsymbol{s}$.

An *action* denotes a sequence of applications of one or more control laws which can be initiated at a locally distinctive state, and terminates after application of a hill-climbing control law with the agent at another distinctive state. A typical action might consist of an open-loop trajectory-following control law to escape from the current neighborhood, then a closed-loop trajectory-following control law to reach a new neighborhood, and finally a hill-climbing control law to reach a new distinctive state.

A *schema* is a tuple $\langle V, A, V' \rangle$, representing the temporally extended event in which the agent takes a particular action A, starting with view V and terminating with view V'. A *routine* is a set of schemas, indexed by initial view.

The schema $\langle V, A, V' \rangle$ has two meanings, declarative and procedural.

$$\text{declarative: } holds(V, s_0) \wedge holds\big(V', result(A, s_0)\big)$$

$$\text{procedural: } holds(V, now) \Rightarrow do(A, now).$$

The relation $holds(V, s_0)$ means that the view V is observed in situation s_0; $do(A, s_0)$ means that action A is initiated in situation s_0; and $result(A, s_0)$ denotes the situation resulting after action A is initiated in situation s_0 and terminates at a new distinctive state.

The declarative meaning is expressed in situation calculus [70]. The procedural meaning is formalized as a guarded command [17] but can equally well be thought of as a rule or a stimulus-response pair.

If the agent's sensory system is poor, or if the environment has few distinguishing features, views sensed at different locations in the environment may not be distinguishable by the agent itself. The ambiguity raised by multiple schemas headed by the same view is minimized by grouping schemas into routines, but its effects cannot be altogether eliminated. To handle adequately the problem of indistinguishable views at different places, we must extend the ontology dramatically, to a topological map describing the structure of the external spatial environment that gives rise to these sensorimotor experiences.

3.2. Turns and travels

At the causal level, it is convenient to classify actions into two categories: turns and travels. Although one can construct abstract environments and/or robots with restricted sensorimotor systems for which these categories break down, they are not problematical for ordinary office environments or street networks.

Once the ontology of the topological level has provided us with places and paths, it is easy to define turns and travels: "A *turn* is an action that leaves the agent at the same place. A *travel* takes the agent from one place to another". Purely within the causal and control ontologies, there are subtler criteria for distinguishing turns from travels that exploit properties of the agent's environment and sensorimotor system. For example, in an otherwise static world, repeating the same turn action gives a periodic sequence of views, while repeating the same travel action typically will not. For another example, if the agent has array-structured sensors such as a sonar ring or a visual image, qualitative properties of sensory flow during motion can distinguish turns and travels [81].

For our purposes in this paper, we will assume that a causal level action description specifies a sequence of control laws and their appropriateness conditions, a classification as turn or travel, and a metrical summary of the net effect of the action from internal effort sensors such as odometry.

- `(Turn` α`)`, where α describes the angle of rotation;
- `(Travel` δ $\Delta\theta$`)`, where δ describes the distance travelled and $\Delta\theta$ describes the net change in orientation.

Partial metrical knowledge is expressed as intervals. Travel along a straight street would give $\Delta\theta = [0, 0]$, while travel along a twisty mountain road might give $\Delta\theta = [-180°, +180°]$. Since an action must begin and end at a locally distinctive state, not every magnitude of turn or travel is a meaningful action.

A robot with a compass sense or a panoramic visual sensor may be able to execute a turn action as a hill-climbing control law. A robot with a more restricted sensory system might need to treat rotation as a trajectory-following control law, requiring a separate hill-climbing step to align the robot with the desired final orientation. In either case, the sensitivity to the local environment provided by the hill-climbing control law gives the "Turn right" action substantially more flexibility and robustness than a "*rotate*$(+90°)$" command.

3.3. Routines

A *routine* is a set of schemas, indexed by initial view. It represents the sequence of actions and intermediate views in a behavior that moves the agent from an initial to a final distinctive state. Fig. 8(b) shows the routine created by exploration of a route through a simulated environment. A routine can be used either as a description of the behavior, or as a procedure for reproducing it.

The basic schema $\langle V, A, V'\rangle$ may be augmented with a goal view G to represent the ultimate destination of a routine. This was used in [50] to explain "capture errors" in route-following.

In order for a complete schema $\langle V, A, V'\rangle$ to be created from observations during behavior, the partially filled schema $\langle V, A, \mathit{nil}\rangle$ must be preserved in working memory during the time required to complete the action A. In case of interruption, it may be that only the partial schema is stored in long-term memory. The partially filled schema $\langle V, A, \mathit{nil}\rangle$ lacks the declarative meaning of the complete schema, but retains its procedural meaning:

$$\text{procedural: } \mathit{holds}(V, \mathit{now}) \Rightarrow \mathit{do}(A, \mathit{now}).$$

A set of complete schemas allows the route to be followed or described, using the *result* component V' as the forward pointer in a linked list, to retrieve the schema describing the next action and its result. If some or all *result* components are missing, the route can still be followed, but only within the physical environment, where the procedural meaning specifies the action to take. The environment then produces the result of the action, allowing retrieval of the next schema. This level of performance accounts for a common state of incomplete knowledge of a route, often described by, "I could take you there, but I can't tell you how".

Consider the alternating sequence of views and actions

$$V_0, A_0, V_1, A_1, V_2, \ldots, V_{n-1}, A_{n-1}, V_n$$

experienced by the agent when travelling along a particular route.

- A routine R is *complete* from view V_0 to V_n if R contains the schema $\langle V_i, A_i, V_{i+1}\rangle$ for each i from 0 to $n-1$.
- A routine R is *adequate* from V_0 to V_n if R contains either $\langle V_i, A_i, V_{i+1}\rangle$ or $\langle V_i, A_i, nil\rangle$ for each i from 0 to $n-1$.

An adequate routine supports "situated action": physical travel from view V_0 to V_n within the environment [1]. It also generalizes naturally to causal graphs such as *universal plans* [87], which are sets of rules specifying the actions to take at *each* state in a state-space to move toward a given goal. In addition to situated action, a complete routine supports cognitive operations such as mental review or verbal description of the route in the absence of the environment.

A *stochastic automaton* is one where a given action in a given state may result in one of several next states, perhaps with probabilities associated with the different transitions. Even in the cognitive map domain, if the sensors are error-prone and the termination condition for the trajectory-following control law is subtle, it may be possible to miss a distinctive state, so that the action terminates at a different distinctive state than expected. Dean, et al. [13,14], use stochastic automata to describe a robot's incomplete knowledge of a spatial environment. Schemas and routines could be extended to express non-deterministic actions and stochastic automata by allowing multiple schemas with the same context and action values, and probabilities indicating the likelihood of each transition.

3.4. Local maps and view-graphs

PLAN [9] is a multiple-representation theory of cognitive mapping with a close relationship to the SSH. PLAN's "local maps" provide an initial bridge between large-scale and visual space representations. The purpose of the local map is to represent the location of visual landmarks to infer the change of orientation needed to view a particular landmark. Resolution may be quite coarse, since visual search will find the landmark if the initial head orientation is reasonably close.

The local map decomposes the egocentric radial directions into a few qualitative classes to the front and sides of the traveller. Each qualitative orientation is associated with a low-resolution 2D image array which represents the approximate positions of objects. Horizontal position in the image improves the resolution of radial direction, and vertical position provides some ordering constraints on distances (assuming a slightly elevated viewing position).

Exploration accumulates local maps and links them into a topological map. Thus, in PLAN, the global metrical map is simply a larger-scale generalization of the local map, representing the environment as if seen from a particular elevated vantage point. The "image" structures in the global map may include objects that are not simultaneously visible, but whose relationships are inferred from the topological and local maps. Thus, the role of local metrical maps in SSH is strongly inspired by local maps in PLAN.

The *view-graph* [23,28,86] is an alternative to a collection of routines as a representation for the schemas stored in memory. Each view is a node in the view-graph and an arc between two nodes V and V' corresponds to the action A in a schema $\langle V, A, V'\rangle$. If views at different places are always distinguishable, the view-graph is closely related to the topological map (or place-graph). On the other hand, when views at different places are indistinguishable, the view-graph structure fails to capture relevant context information that is embodied in a routine. Notwithstanding this representational difference, research based on view-graphs is highly relevant to the SSH causal level.

Gillner and Mallot [28] studied cognitive mapping by human subjects exploring a large-scale virtual reality environment (Hexatown). Their subjects were clearly able to build effective cognitive maps from experience with purely visual information. The states of knowledge they observed were much more consistent with a graph-like representation connecting local elements than with a global "view from above" metrical map. However, their results could not distinguish between a graph of places and a graph of views: some experiments strongly supported the view-graph representation, while others were more compatible with the place-graph.

From the perspective of the SSH, their results support the validity of three claims. First, the simple view-action interface between the control level and the causal and topological levels is compatible with what we know of human spatial behavior. Second, the causal and topological graphs are more fundamental than the global metrical map, in humans at least, when building a cognitive map from observations. Third, *both* the causal level (view-graph) and the topological level (place-graph) are present in the cognitive map.

Franz et al. [23], describe experiments with both physical and simulated robot systems demonstrating the creation of view-graphs of irregular environments using information from a panoramic visual sensor. The only action is "homing" (i.e., hill-climbing) between two similar visual images: moving so as to transform the current visual image toward the stored image from the destination. When exploring unknown territory, a new view is added to the graph whenever similarity of the current image to the previous stored view becomes less than some threshold. Since views are defined by threshold dissimilarity, the view-graph is a relatively high-resolution description of the environment. By contrast, in the SSH views are defined by distinctiveness, so they are separated by the larger distances that trajectory-following control laws can cover.

4. The topological level

The topological map describes the environment as a collection of places, paths and regions, linked by topological relations such as connectivity, order, boundary and containment. Places, paths and boundary regions are created from experience represented as a sequence of views and actions. They are created by *abduction*, positing the minimal additional set of places, paths, and regions required to explain the sequence of observed views and actions.

- A *place* describes part of the environment as a zero-dimensional point. A place may lie on zero or more paths. A place may also be defined as the abstraction of a region. A turn action leaves the agent at the same place.
- A *path* describes part of the environment, for example a street in a city, as a one-dimensional subspace. The two directions along a path are $dir = +1$ and $dir = -1$. A travel action takes the agent from one place to another along a single path. A path

may describe an order relation on the places it contains, and it may serve as a boundary for one or more regions.

- A *region* represents a two-dimensional subset of the environment. A region may be defined by one or more boundaries, by a common frame of reference, or by its use in an abstraction relation.

4.1. Topological relations

The topological relations represent location of views, connection of places and paths, order of places on paths, and boundaries and membership of regions.

$at(view, place)$	*view* is seen at *place*
$along(view, path, dir)$	*view* is seen along *path* in direction *dir*
$on(place, path)$	*place* is on *path*
$order(path, place1, place2, dir)$	the order on *path* from *place1* to *place2* is *dir*.
$right\_of(path, dir, region)$	*path*, facing direction *dir*, has *region*
$left\_of(path, dir, region)$	on its right (respectively left)
$in(place, region)$	*place* is in *region*.

The following axioms assert that each path maintains a partial order of the places on it. (A, B and C are places, and free variables are universally quantified.)

$$order(path, A, B, dir) \rightarrow on(A, path) \wedge on(B, path), \tag{10}$$

$$\neg order(path, A, A, dir), \tag{11}$$

$$order(path, A, B, +1) \leftrightarrow order(path, B, A, -1), \tag{12}$$

$$order(path, A, B, dir) \wedge order(path, B, C, dir) \rightarrow order(path, A, C, dir). \tag{13}$$

In many domains, it also makes sense to require the existence of a "turn around" action.

$$\exists \alpha [along(V, path, dir) \wedge \langle V, (\mathtt{turn}\ \alpha), V' \rangle \wedge along(V', path, -dir)]. \tag{14}$$

4.2. Abduction to places and paths from views and actions

We use the following observations as the basis for abduction of the connectivity properties of places and paths, given schemas relating views and actions. The order of places along the current path is inferred from a travel action and the current direction.

- Every view is observed at a place.

$$\forall view\, \exists place\; at(view, place). \tag{15}$$

- A turn action leaves the traveller at the same place.

$$\langle V, (\mathtt{turn}\ \alpha), V' \rangle \rightarrow \exists place[at(V, place) \wedge at(V', place)]. \tag{16}$$

- A travel action takes the traveller from one place to another on the same path, facing the same direction.

$$\langle V, (\mathtt{travel}\ \delta), V' \rangle \wedge \delta \neq 0 \rightarrow \exists p_1, p_2[p_1 \neq p_2 \wedge at(V, p_1) \wedge at(V', p_2)], \tag{17}$$

$$\langle V, (\texttt{travel}\,\delta), V'\rangle \\ \rightarrow \exists path, dir[along(V, path, dir) \wedge along(V', path, dir)], \tag{18}$$

$$\langle V, (\texttt{travel}\,\delta), V'\rangle \wedge \delta \neq 0 \\ \rightarrow \exists p_1, p_2, path, dir[at(V, p_1) \wedge at(V', p_2) \wedge along(V, path, dir) \wedge \\ order(path, p_1, p_2, dir)]. \tag{19}$$

The abduction consists of binding constants to the existentially quantified variables in the above axioms. It is implemented by rules that search for known places and paths with the required properties, creating new constants if existing ones cannot be found.

Levitt and Lawton's QUALNAV [58,64] provides an interesting comparison with the SSH. The traveller is assumed to have a panoramic visual sense for distant landmarks. It crosses a qualitative spatial boundary when it is colinear with two distant landmarks. The regions defined by these boundaries are the qualitative neighborhoods. QUALNAV does not define individual distinctive states within these neighborhoods, but defines the topological map as the adjacency graph of the neighborhoods. The control law to follow a topological edge is specified in a natural way as bisecting the angle between the two landmarks defining the boundary that must be crossed to pass from one neighborhood to the next. Thus, although it is restricted to environments where distant landmarks are visible, QUALNAV shares many characteristics with the SSH framework, because it abstracts continuous space to qualitatively uniform regions and because topological links are defined in terms of continuous control laws.

Mataric [69] also built a topological map whose elements were characterized by the control laws used to traverse them. However, her "places" are defined by trajectory-following control laws, and so correspond most closely to actions or paths in the SSH. Furthermore, the adjacency relations among "places" are the only structures in her representation that correspond to places in the SSH.

4.3. *Regions, boundaries and abstraction*

Regions are sets of places, grouped together because they lie on one side of a certain boundary; because they share a certain 2-D metrical frame of reference; or because they are abstracted to the same place in a higher-level topological map.

A directed path divides the world into two regions: one on the right and the other on the left. A *bounded region* is defined by a directed path with the region on its right or on its left.

$$right\_of(path, dir, region) \leftrightarrow left\_of(path, -dir, region) \tag{20}$$

$$\neg[right\_of(path, dir, region) \wedge left\_of(path, dir, region)]. \tag{21}$$

Membership of places in bounded regions can be incrementally acquired during travel by a rule such as: *If the agent travels along a certain directed path, turns right, then travels again to reach a certain place, then that place lies within the region right_of that directed path.* Formalising this rule as an axiom, given the definitions and axioms we already have, is straight-forward but a bit tedious.

$$
\begin{aligned}
&\langle V_1, (\texttt{travel}\ \delta_1), V_2\rangle \wedge \\
&\langle V_2, (\texttt{turn}\ \alpha), V_3\rangle \wedge \\
&\langle V_3, (\texttt{travel}\ \delta_2), V_4\rangle \wedge \\
&0 < \alpha < 180^\circ \\
&\rightarrow \\
&\exists path_1, path_2, dir_1, dir_2, P_1, P_2, P_3, R[\\
&along(V_1, path_1, dir_1) \wedge along(V_2, path_1, dir_1) \wedge \\
&along(V_3, path_2, dir_2) \wedge along(V_4, path_2, dir_2) \wedge \\
&at(V_1, P_1) \wedge at(V_2, P_2) \wedge at(V_3, P_2) \wedge at(V_4, P_3) \wedge \\
&right\_of(path_1, dir_1, R) \wedge \\
&in(P_3, R)].
\end{aligned}
\tag{22}
$$

One can use containment relations among the *right_of* and *left_of* regions to define when two paths are topologically parallel, which in turn generalizes to topological grid structures for a larger region. Both the topological grid structure and the simpler boundary relations are useful for finding subgoals during route-finding.

Local metrical frames of reference can be propagated from one place to its neighbors along a path segment in response to a travel action $(\texttt{travel}\ \delta\ \Delta\theta)$ if $\Delta\theta \approx [0, 0]$. An *orientation region* can be defined for the set of places using the same reference frame.

An *abstraction region* represents the set of places in a detailed map that is abstracted to a particular place in a larger granularity map. For example, a large granularity map of central Texas might represent I-35 as a path linking three places, Dallas, Austin and San Antonio. Such a hierarchical topological map is clearly useful for finding routes in a large graph, though its structure may make it difficult to find *optimal* routes between arbitrary places.

To use an abstraction hierarchy to find a usable route requires upward and downward mappings in the hierarchy [45,46].

– **Upward Mapping:** a place at a lower level is mapped to the place corresponding to the abstraction region that contains it.
– **Downward Mapping:** a $\langle place, path, dir\rangle$ tuple at the higher level is mapped to a corresponding $\langle place, path, dir\rangle$ tuple at the lower level.

The downward mapping is more complex than the upward mapping to reduce the inevitable ambiguity of inverting an abstraction relation. It is inspired by the relation between a limited-access highway and the network of surface streets.

Although the TOUR model includes a representation for this abstraction hierarchy [45, 46], there is as yet no theory of how the hierarchy is acquired.

4.4. The TOUR model

The TOUR model [45–47] structures the abduction as an incremental, opportunistic, spatially localized computation. It is organized around a set of fluents (terms with time-varying truth values) representing the traveller's current view, action, schema, routine,

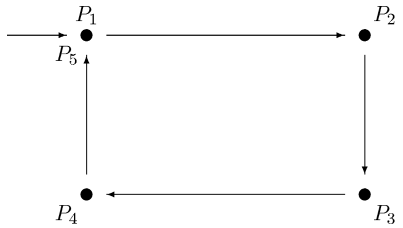


Fig. 7. Exploring "around the block" raises several problems: recognizing the same place under different names, and cumulative position error.

place, path, direction, region, frame of reference, and heading. Collectively, these fluents are called the "You-Are-Here pointer".

The input to the TOUR machine [5] is the alternating sequence of views and actions

$$V_0, A_0, V_1, A_1, \ldots, V_{n-1}, A_{n-1}, V_n$$

experienced by the traveller (human or robot) as it moves through the environment.

The state of the TOUR machine is only defined when the SSH control level has brought the traveller to a distinctive state. The current view is provided by the agent's perceptual system. The current action is provided by the agent's decision to invoke a particular action, whether in response to a self-generated plan or an external instruction. Temporal continuity is provided by storing the previous view in working memory as well as the current view.

The heart of the abduction process is "find-or-create retrieval". Immediately after an action has terminated, a new current view V' is provided by the perceptual system, but the values of the other components of the You-Are-Here pointer are unknown. If the value of the current place cannot be inferred from the information already in the cognitive map, a new place constant is created by instantiating the existentially quantified variables in Eqs. (15), (16) or (17). Similarly with paths and Eqs. (18) and (19). When enough information about the current situation is present in the You-Are-Here pointer, the TOUR machine can deduce time-independent assertions of topological or local metrical relations, using axioms such as the following.

$$\mathit{current\_place}(p) \wedge \mathit{current\_view}(v) \rightarrow \mathit{at}(v, p), \tag{23}$$

$$\mathit{current\_path}(p) \wedge \mathit{current\_direction}(d) \wedge \mathit{current\_view}(v) \rightarrow \mathit{along}(v, p, d), \tag{24}$$

$$\mathit{current\_place}(p) \wedge \mathit{current\_path}(\mathit{path}) \rightarrow \mathit{on}(p, \mathit{path}). \tag{25}$$

As the cognitive map becomes more richly specified, similar rules can deduce values for fluents in the "You-Are-Here" pointer, for example inferring the current path or place from local metrical information and the magnitude of a turn or travel action.

The hard part of the abduction, and the part most likely to be deductively unsound, is to determine when two existing constants refer to the same place or path in the environment.

For example, suppose the explorer goes around a rectangular block in a new environment (Fig. 7). The TOUR machine incrementally creates place and path descriptions for the

[5] The set of rules implementing the abduction is called the "TOUR machine", leading the author down the slippery slope to a demonstration and evaluation function called the "TOURing test".

corners of the block. Place description P_5 is created to represent the destination of the travel action from P_4 before doing the more expensive inference required to conclude that $P_5 = P_1$. This is a feature, not a bug, since it preserves the incremental, interruption-tolerant character of TOUR machine inference, albeit leaving the topological map in a weakened form when it fails to identify self-intersection points.

The "rehearsal strategy" [55,56] tests the hypothesis that $P_5 = P_1$ by using physical exploration starting from P_5 to check for the neighbors predicted for P_1. This strategy, limited to any finite radius, is clearly vulnerable to false positive conclusions. However, Dudek, et al. [19] show that correct qualitative strategies exist if the traveller can leave a uniquely recognizable mark on the environment, or if there is a reachable, uniquely recognizable, "home base". Local or global metrical information, even when incomplete, is clearly helpful in testing identity hypotheses.

From the implementation perspective, it is relatively easy to handle the case of a newly created "place-holder" symbol like P_5, about which very little is known. If the identity $P_5 = P_1$ is quickly learned, all occurrences of P_5 can simply be replaced by P_1. Much less frequently, an identity may be discovered between two richly described places that were believed to be distinct. The implications of such a change can be arbitrarily far-reaching within the knowledge base. The methods for such extensive knowledge reorganization are beyond the scope of this paper.

The abduction rules as currently performed are informally derived from the theory. Remolina and Kuipers [83,84] express the abduction more formally in terms of prioritized circumscription and Lifschitz' nested abnormality theories [65].

4.5. Local 1-D geometry

Some metrical knowledge consists of quantitative attributes that can be treated as annotations on the symbolic framework provided by the topological map. For these attributes, the dependence of the quantitative knowledge on the topological map is clear: until places and paths have been created, there are no objects for these quantities to be attributes of.

Observations of the magnitudes of actions provide information about the local geometry of places and paths. $\langle V, \texttt{(travel } \delta\texttt{)}, V'\rangle$ provides evidence about the distance between two places on the current path. $\langle V, \texttt{(turn } \alpha\texttt{)}, V'\rangle$ provides evidence about the angle between obstacles and/or paths at the current place. This information can be represented as 1-D (linear or circular) metrical properties of the individual places and paths in the topological map. These properties are accumulated incrementally by the same abductive process that builds the topological map.

Geometric information local to a particular place or path is represented by two predicates:

- $\mathit{radial}(\mathit{place}, \mathit{view}, \mathit{heading})$. When the agent is located at *place*, *view* is obtained when facing in the direction *heading*, which is a quantity representing the clockwise angle from the zero heading of a frame of reference local to that place.
- $\mathit{position1}(\mathit{path}, \mathit{place}, \mathit{position})$. When the agent is located at *place*, on *path*, *position* is its one-dimensional coordinate with respect to a frame of reference local to *path*.

Given these predicates, we can define the relations among metrical observations at the causal level, and local geometry at the metrical level.

- From $(\texttt{turn}\ \alpha)$:

$$
\begin{aligned}
&\langle V_1, (\texttt{turn}\ \alpha), V_2\rangle \\
&\rightarrow \\
&\exists P, h_1, h_2[\\
&radial(P, V_1, h_1) \wedge \\
&radial(P, V_2, h_2) \wedge \\
&h_2 = h_1 + \alpha \bmod 360^\circ].
\end{aligned}
$$

- From $(\texttt{travel}\ \delta)$:

$$
\begin{aligned}
&\langle V_1, (\texttt{travel}\ \delta), V_2\rangle \wedge d \neq 0 \\
&\rightarrow \\
&\exists P_1, P_2, path, dir, pos_1, pos_2[\\
&at(V_1, P_1) \wedge at(V_2, P_2) \wedge \\
&along(V_1, path, dir) \wedge \\
&position1(path, P_1, pos_1) \wedge \\
&position1(path, P_2, pos_2) \wedge \\
&pos_2 = pos_1 + \delta \cdot dir].
\end{aligned}
$$

The abductive rules that implement these axioms have the capability to define initial zero headings and positions when no previous local frame of reference is known. 1-D position is an interval quantity: zero is an arbitrary landmark.

4.6. Way-finding

The topological level of representation is a graph of places, paths and regions. It supports a variety of problem-solving methods that can exploit different types of knowledge that might be available (cf. [20,35]).

- The graph of places and paths can always be searched blindly. Where distance estimates exist, one can use A* search [75] or Dijkstra's algorithm [16].
- Where direction and heading estimates exist, heuristic search can prefer motion in the direction of the goal.
- Where boundary region information exists, and a desired route must cross a region boundary, then any place on that boundary is a potential subgoal. Collections of related boundaries can be organized into structures such as topological grids that provide a selection of potential subgoals [45,46].

- Where an abstraction region hierarchy exists, a complex way-finding problem can be abstracted to a simpler high-level problem, plus simpler connection problems at the endpoints [45,46].

5. Global metrical mapping

A global 2-D analog metrical representation allows powerful metrical inferences, but can be expensive and error-prone to create, especially from local observations during travel.

5.1. Single global frame of reference

The "Map in the Head"—a globally consistent 2-D analog representation—is a popular theory of spatial knowledge, even though its limitations as a cognitive theory have long been recognized [18,48,67].

Many robot mapping projects [7,8,11,63,72,94] take as their goal the construction of a map that specifies the locations of the robot and relevant environmental features within a single, global frame of reference. This has natural appeal: when an agent knows its position and orientation within a global map, it is straight-forward to infer the vector to any known feature of the environment. Furthermore, in an environment subject to *perceptual aliasing*—many places appearing similar or identical, as in the desert, in the woods, or on the surface of Mars—accurate localization in a global frame of reference can compensate for not being able to tell the places apart.

The first problem with global metrical mapping is that an exploring agent, robotic or human, has useful states of knowledge that are not expressible as coordinates within a single frame of [48]. In particular, during travel, any agent will accumulate position uncertainty with respect to a global frame of reference tied to its initial position. For a convex region, or more generally a simply-connected region, a suitable exploration strategy can keep these cumulative errors within bounds. However, when the region is not simply connected, traveling "around the block" and closing the loop (Fig. 7) raises the difficult problem of matching the current position to the original frame of reference [7,8].

Experience suggests that orientation error is more common, and its impact accumulates more seriously, than translation error. The phenomenon of "walking in circles" when lost in the woods suggests that cumulative orientation error is a serious problem for humans, as well as robots. Some animals such as insects and birds have polarized light or magnetic field sensors that can be used as a compass sense, giving greatly improved dead reckoning abilities [24].

The second problem with global metrical mapping is the space and time cost of the mapping algorithm. A uniform 2-D occupancy grid representation [72,73,94] requires a number of cells that is quadratic in the diameter of the environment divided by the length of the grid cell. Hierarchical structuring methods such as partitioning office environments into rooms can reduce the cost of these algorithms, but in many cases these methods amount to implicitly identifying and exploiting the topological structure of the environment that is explicitly represented by the SSH.

The feature-mapping representations used by Leonard and Durrant-Whyte [63] identifies environmental features by their sensory signatures and assigns them locations in the single global frame of reference. The cost of this representation is proportional to the number of features in the environment and its resolution is not limited by grid cell size. However, it shares the problem of cumulative error, and it depends on having sensory models for the different environmental features it is to recognize, making it less robust in arbitrary environments than occupancy grids.

5.2. *Patchwork mapping*

Another way to build a global metrical map is to create a loosely-coupled collection of local "patch" maps of simply-connected neighborhoods, each with its own local frame of reference. The relationships linking the patches may be purely qualitative, of they may have imprecise quantitative information associated with them. Fig. 2(c) shows the result of relaxing local maps of place neighborhoods and path segments into a single frame of reference, under the assumption that the distortions required to map the local headings and distances into a planar map are evenly distributed across all the patches [56].

This approach draws strongly on the work of McDermott and Davis [71], who proposed a very flexible representation for metrical knowledge consisting of multiple frames of reference. It used real-valued intervals to express uncertainty both of locations within a single frame of reference and of the relation between different frames of reference.

Another approach to constructing a global metrical map from local patches focuses on real and constructed visual images rather than 2-D analog spatial representations. PLAN [9] represents the local metrical map as a collection of coordinated visual images, and the global metrical map is a constructed "image" of the environment as it might be seen from a distant vantage point.

Thrun et al. [95] demonstrated the value of a topological framework for reducing cumulative estimated position error when constructing a global metrical map. This method augments the occupancy grid with a set of widely spaced "significant places" that are indicated to the robot by a human operator. The localization-mapping cycle is applied first only to the set of significant places, representing the large-scale structure of the environment, then to the occupancy grid representing the fine structure. This hybrid topological-metrical mapping method resists cumulative position error, converges faster, and achieves high accuracy.

Accurate local metrical maps can be important to local control laws, and they are relatively easy to construct from the information available from vision, laser range-finders, and even sonar. Global metrical information is useful when it is available, but in sufficiently perceptually rich environments, it is seldom on the critical path for exploration, map-learning, route-planning or navigation. Global metrical mapping is essential only when few places are perceptually distinctive, and it is successful only when position uncertainty can be prevented from accumulating.

6. Evaluation by robot implementation

A working implementation is a demonstration that the different representations can function together coherently and effectively. It also demonstrates qualitative properties of the system.

6.1. Simulated agent in continuous office environment

Kuipers and Byun [55,56] implemented the control, topological, and metrical levels on a simulated robot with a radial array of 16 range-sensors subject to both random and systematic errors similar to those of the Polaroid sonar sensor.

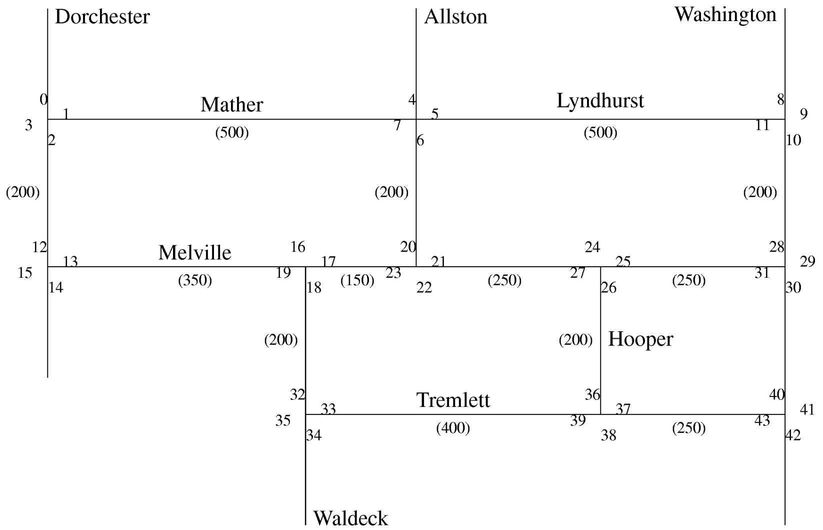
Fig. 2 shows the trace of the NX robot's exploration of its simulated environment, a fragment of the topological map constructed from that exploration, and a representation of the patchwork of local metrical maps, relaxed into a single frame of reference for display. Conclusions to be drawn from this experiment are:

- In this office environment, NX identifies a concise set of locally distinctive topological map elements (20 places and 23 edges), corresponding well with the points a human viewer considers distinctive.
- Random sensor error is well handled by the control laws. Systematic sensor error modeled on the specular reflection errors of sonar sensors, are handled effectively (if perhaps tediously) by approaching suspected illusory open spaces for closer inspection.
- Perceptual aliasing (distinct places with identical local sensory signatures) is effectively disambiguated through physical exploration guided by the rehearsal strategy.
- An exploration agenda consisting of unexplored edge-ends effectively drives exploration of the environment and construction of a complete topological map, and provides a restricted set of candidates for disambiguating cases of perceptual aliasing.
- Routes constructed at the topological level can be effectively mapped down to the control level and executed.
- Exploration, topological map-building, way-finding and navigation can all be done effectively without the use of quantitative information outside of the local control laws themselves.

6.2. Simulated agent in discrete urban environment

The TOUR model consists of the symbolic representation and inference portions of the SSH: the causal and topological levels, without the control level or local or global 2-D geometrical maps. Since its original incarnation [45], the TOUR model has been implemented numerous times, building frame-structured descriptions of places, paths, and the other aspects of the SSH causal, topological, and 1-D metrical levels. The original was implemented in MacLisp, and the author reimplemented it first in an ad hoc rule language embedded in Lisp, and later in the logic-programming language Algernon [12]. Results from the Algernon implementation are shown in Fig. 8. Larger subsets of the SSH, augmenting the symbolic TOUR model with the control level, have been used to build cognitive maps for simulated robots by Byun [55,56] and by Pierce [81], and for a physical robot by Lee [62].

Fig. 8(a) shows a simulated portion of an urban environment. The environment is richly enough structured and perceived to avoid perceptual aliasing. The agent is taken on a tour of the neighborhood, going around each of the four blocks once in each direction, traveling along each path segment, visiting each place and observing each view. Of the 44 views in



(a)

```
< V5 TRAVEL6[500] V9 >
< V9 TURN17[90] V10 >
< V10 TRAVEL7[200] V30 >
< V30 TURN18[90] V31 >
< V31 TRAVEL8[500] V23 >
< V23 TURN19[90] V20 >
< V20 TRAVEL9[200] V4 >
< V4 TURN20[90] V5 >
< V5 TRAVEL10[500] V9 >
```

(b)

```
Place-1:
  Isa: places abstract-objects objects things
  View-at: v9 v8 v10 v11
  On-dpath: dpath-4 dpath-10 dpath-1 dpath-13
  View-angle: (v9 v8 -90) (v8 v9 90) (v9 v10 90)
              (v8 v11 -90) (v10 v11 90)
  Local-heading: (v9 0) (v8 270) (v10 90) (v11 180)
  On-path: path-2 path-3

Path-2:
  Isa: paths abstract-objects objects things
  Place-on2: place-1 place-2 place-9
  Position: (place-2 0) (place-1 200) (place-9 -200)
  Dpath: (dpath-10 pos) (dpath-4 neg)
  Order: (place-2 place-1) (place-9 place-2)
```

(c)

Fig. 8. TOUR model exploration. (a) Map of a simulated portion of Dorchester, Massachusetts. The small numbers around each intersection indicate the views obtained in the four directions. (b) The sequence of causal schemas experienced during a route around a block. (c) Frames for a place and a path after the exploration is complete. (The slot-names in this implementation are not the same as the relation-names used in this paper.)

the environment, 9 are experienced only once, 23 are experienced twice, 10 are seen three times, and only 2 are seen four times. Fig. 8(b) shows a causal route description constructed after part of the tour, and Fig. 8(c) shows two of the frames constructed, describing a particular place and a particular path. Conclusions to draw:

- With an appropriate ontology, an opportunistic approach to deductive and abductive inference efficiently learns the topological structure of an environment.
- In the absence of perceptual aliasing, abductive reasoning is simply and successfully implemented as “find-or-create retrieval” of frames in the knowledge base.

6.3. Physical robot in office environment

The pitfall of robotics in simulated environments is the possibility that successful performance could be due the simplified environment provided by the simulator, rather than due to the proposed method. To evaluate this possibility, Lee [62] demonstrated a successful implementation of the control level on a sonar sensing robot (RWI B12) in an office environment, abstracting its behavior to simplified causal and topological representations.

With physical sonar, specular reflections could make important obstacles simply invisible. To overcome this problem, it was important to implement a local observer process that built a local metrical map of nearby obstacles when they were easy to perceive. The local map would then support "virtual sensing" of the object when it became difficult or impossible to perceive directly.

Fig. 9 demonstrates the robot's behavior as it detects the termination of a trajectory-following control law and hill-climbs to a distinctive state. Several other useful methods were developed and demonstrated, including a phase-diagram for selecting trajectory-following control laws (Fig. 11) and a decision tree for classifying the structure of the local place neighborhood and selecting hill-climbing control laws.

- With a few modifications, the SSH exploration, mapping and navigation methods extend from simulated to physical robots.

While experimental results are not yet available, our laboratory is using the SSH as the knowledge representation foundation for an intelligent wheelchair [32]. This robot uses laser range sensors and binocular vision. Sonar sensors will be added, but used only as touchless bump sensors.

6.4. Simulated learning agent in unknown continuous sensorimotor environment

Pierce and Kuipers [81] demonstrated a hierarchical learning system for a robot with initially uninterpreted sensors and effectors. By analyzing the effects of its actions on its sensor input, the agent learns a hierarchy of representations of its sensorimotor system, leading up to the SSH control level. The agent learns: the structure of its sensor set; the effect of its actions on its sensors; a useful set of primitive actions; a useful set of higher-level features defined in terms of primitive sensory features; local state variables to characterize the current environment as described by the features; control laws for changing one local state variable while holding the others fixed; and criteria for defining distinctive states. Fig. 10 shows the patterns of exploration through a simple environment at three stages of the learning process:

(a) random motion to learn useful sensory and motor features;
(b) open-loop wall-following control laws; and
(c) closed-loop wall-following control laws.

- It is possible for an agent to build a cognitive map in the SSH representation, even without prior knowledge of the properties of its sensors and effectors.

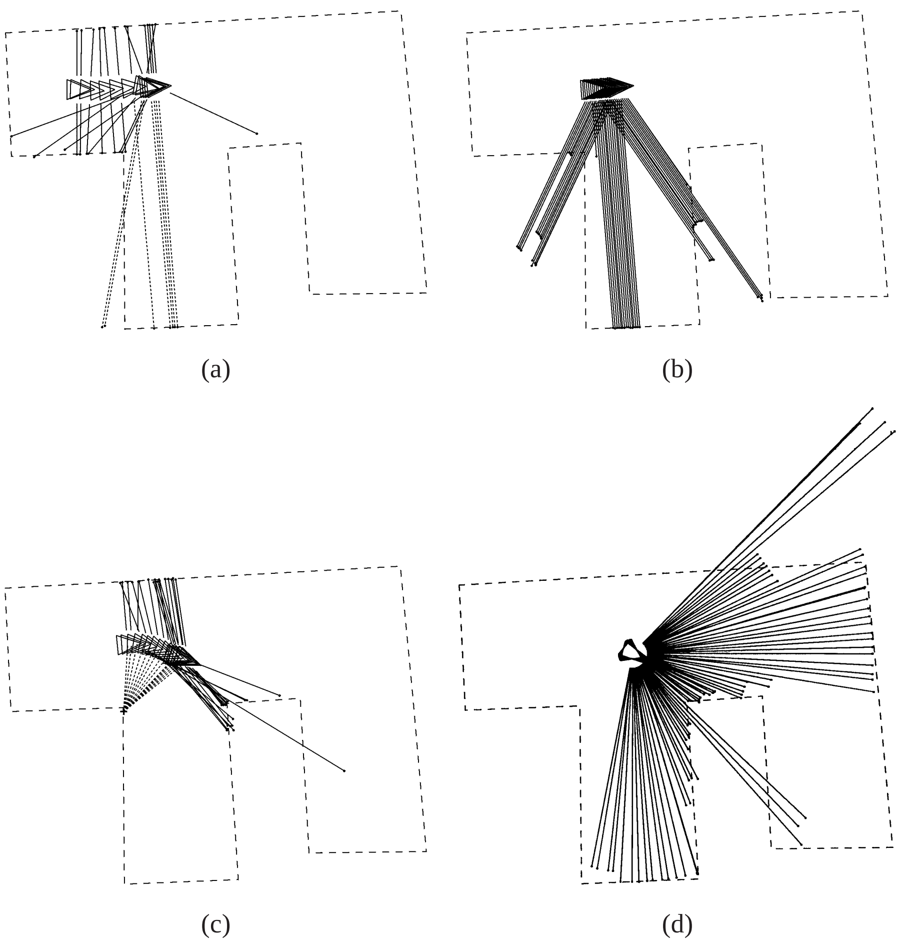


Fig. 9. Control behaviors leading Spot (RWI B12) to a distinctive state. (a) Spot is following the midline when it detects sudden increase in distance to right wall. (b) Moving slowly, Spot localizes the convex corner in a local frame of reference. (c) Using "virtual sensing" of the convex corner, Spot follows a parabolic trajectory equidistant from corner and wall, until reaching a point the same distance from the forward obstacle. (d) Spot rotates to face directly towards the forward obstacle, defining the distinctive state.

7. Applying the spatial semantic hierarchy

To implement the SSH on a given robot, it is necessary to define its connection to the robot's particular sensors and effectors. This requires the robot's designer to define trajectory-following and hill-climbing control laws, to define views, and to show how the local metrical map is derived from sensory input. The SSH has been formulated to avoid dependence on the properties of particular sets of sensors or effectors, but rather to depend on the controllability of the dynamical system consisting of the agent in its environment.

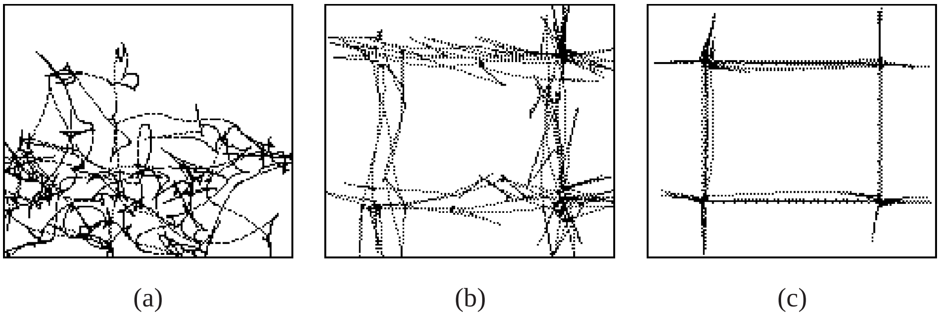


Fig. 10. Exploring a simple world at three levels of competence. Reprinted from Artificial Intelligence, Vol. 92, D.M. Pierce and B.J. Kuipers, Map learning with uninterpreted sensors and effectors, p. 204 (1997), with permission from Elsevier Science.

7.1. Trajectory-following control laws

Trajectory-following control laws, for example wall- or corridor-following, keep two out of the three degrees of freedom in the robot's state near setpoint values, leaving one degree of freedom for forward or backward progress. Individually, such control laws are well understood (e.g., [15,98]).

To implement the SSH control level for a given robot, the designer must specify a set of trajectory-following control laws, along with their conditions of applicability, in forms suitable both for selecting the control law to apply during exploration and for detecting when the control law has reached the end of its region of applicability. As discussed in Section 2.2, the regions of applicability may have fuzzy boundaries, and heterogeneous control laws may be composed from multiple simpler laws. Fig. 11 shows the regions of applicability of four trajectory-following control laws appropriate for range-sensing in an office environment, based on [62, Fig. 4.8].

Vision sensors support a different set of control laws, for example homing on a visually distinctive target, homing on the vanishing point of dominant sets of line-segments in the image, or aiming toward the midpoints of line segments defined by objects to pass on the left and on the right. Just as with range sensors, these control laws can be defined as reactive responses to sensory input, prior to constructing a model of the environment. Other trajectory-following control laws suitable for outdoor navigation include visual lane-following for autonomous vehicles [36], crossing qualitative boundaries defined by pairs of distant landmarks [64], and following a fixed heading until a destination is reached [29, 99].

Note that while control laws use continuous sensory input and issue continuous output signals, they embody very little knowledge about what the sensors are sensing or what the effectors are effecting. The *designer* may know the structure of the environment, but the robot does not. If the robot learns its control laws from its own experience, it may be able to build a perfectly functional cognitive map without explicitly or implicitly representing what the world is "really" like [81].

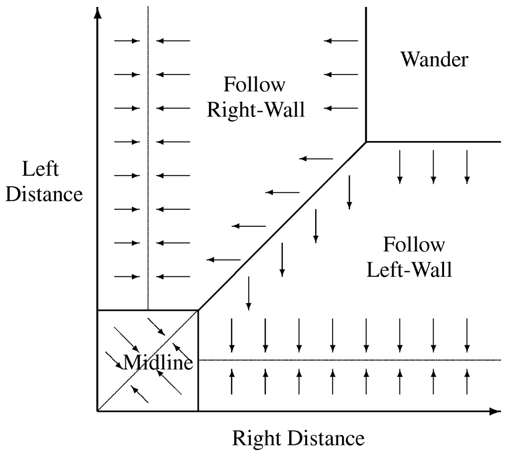


Fig. 11. Selection criteria for trajectory-following control laws. In an office environment with range-sensors, a trajectory-following control law is selected depending on distances to obstacles on the left and on the right. Arrows show the direction of flow in the region appropriate to each control law.

7.2. Hill-climbing control laws

A hill-climbing control law eliminates all three degrees of freedom in the robot's state description: it brings the robot to a locally distinctive state. The designer must specify (or the robot must learn) a set of hill-climbing control laws appropriate to the type of environment it is to function in. For a range-sensing robot in a room-and-corridor environment (e.g., Fig. 2) the most common hill-climbing control law seeks the point equidistant from three nearby obstacles. Another important type of hill-climbing control law seeks the point along a trajectory at which a sudden change takes place, for example the sudden increase in range-sensor values when the robot detects a cross-corridor. In an outdoor environment, a hill-climbing control law can be defined in terms of distance to nearby obstacles or landmarks [56,62], distance from qualitative landmark pair boundaries [64], or estimated position along a trajectory [29,99].

The selection of an appropriate hill-climbing control law starts when the current trajectory-following control law terminates (Fig. 3). The identity of the current control law and the nature of the assumption violation that terminated it are important clues for the selection of the hill-climbing control law, but there may be other important features sensed in the current neighborhood that were not attended to by the trajectory-following control law.

The hill-climbing control law terminates when it has arrived at, or near within some small tolerance, the locally distinctive state in the current neighborhood.

7.3. Local metrical mapping

The local metrical map is a Euclidean coordinate frame of reference representing the state of the agent and the locations of perceived obstacles within the local neighborhood. As discussed in Section 2.5, construction of the local metrical map is not dependent on physical travel to a distinctive state. On the other hand, it does depend on a stronger interpretation of the meaning of sensor input and motor action than is required simply to execute a control law.

A procedure for constructing the local metrical map embodies knowledge of the relationship between the agent's sensors and its effectors, through the intermediate structure of the local frame of reference. The agent must be able to predict the effects of its actions on its state within the local frame of reference, and it must be able to predict the effect of objects nearby on the inputs it receives from its sensors. The local frame of reference provides a conversion factor between motor units and sensor units.

These relationships can be provided by the robot's designer or learned from experience [81]. Once the fundamental relationships are known, the local metrical map can be created by a variety of methods including Kalman filters and Bayesian maximum likelihood estimators [62,63,91,94]. These methods are primarily oriented toward odometry and range-sensing. Other methods have been developed for visual sensing robots [40,61].

7.4. Views

When the agent is at a distinctive state, the current sensory image is abstracted to a *view*. The SSH model of large-scale space treats views as having no internal structure: a view can serve as an index for storage and retrieval of a causal schema, and two views can be matched for identity. This defines a narrow interface between a model of large-scale space and possible models of visual (or other sensory) space.

The robot designer must specify a description to be derived from the sensory image the robot receives at a distinctive state. The SSH causal and topological levels work best if the current view uniquely identifies the current distinctive state. False positive matches (different states with indistinguishable views) require either complex reasoning or physical travel to resolve ambiguities (Section 4.4). False negative matches (the same state not recognized because of different views on different occasions) result in multiple place descriptions created for the same place, possibly requiring substantial reorganization of the map when the identity is recognized. Therefore, the designer must select a trade-off between these two error types when specifying how a view describes the sensory image.

The sensory image of a neighborhood provided by a ring of 12 or 16 sonar sensors is particularly weak due to specular reflections. For this reason, Lee [62] built a local sensory target map of each neighborhood, and defined a view as this local map plus the position and orientation of the robot within it. The effectiveness of image-based views can be improved with higher-quality images from visual sensors or infrared laser range sensors. As our understanding of the relationship between visual space and large-scale space improves, it seems likely that the proper interface will subsume both the view and the local metrical map.

7.5. Symbolic levels

The interaction between the cognitive map and the sensorimotor system of the robot is mostly subsumed by the control laws, the views, and the local metrical map. The symbolic levels of description in the SSH causal, topological, and 1-D metrical levels are largely independent of the specific properties of the robot.

7.6. Exploration strategy

The strategy for exploring the environment is expressed at the topological level, and so is largely independent of the agent's sensorimotor system. However, the strategy to be chosen depends on the pragmatic goals of the agent. The only goal of the NX robot [55, 56] is "curiosity": the need to identify the place connected to any unexplained path-end in its topological map. Whenever NX reaches the place at the end of a path segment, it determines whether that place is a familiar one or a new one, using the rehearsal strategy (Section 4.4) if necessary. For each new place, NX adds the path-ends identified at that place onto an exploration agenda. NX simply explores its small environments exhaustively, until the exploration agenda is empty and the topological map is complete. In larger or unbounded environments, different behaviors would result from different ordering strategies on the exploration agenda: conservative, close-to-home breadth-first search; wide-ranging, weakly connected depth-first search; or prioritized strategies oriented toward other goals such as finding food, avoiding predators, or following tropisms.

A robot vehicle such as an intelligent wheelchair or automobile will not have the autonomy to select travel actions in service of its own exploration goals. However, the TOUR model constructs the causal and topological maps opportunistically, regardless of how the actions are selected. Without the autonomy to perform a rehearsal strategy, ambiguity among places will increase, but sensors such as vision and GPS [6] will reduce its impact.

8. Discussion

8.1. Coping with uncertainty

One of the key challenges of mapping and navigation in large-scale space is coping with uncertainties in sensing and actuation. The SSH decomposes this uncertainty into components that are handled effectively by the different representations.

Another major design goal for the SSH is the ability to support effective action in spite of substantial limitations in sensing, effecting, or computing resources. Therefore, it supports weaker but still useful qualitative methods for navigation when resources are not available to support high-resolution quantitatively-precise mapping.

Traditional control theory [15,22,98] provides powerful methods for achieving reliable, reproducible behavior in spite of significant sensor and effector uncertainty, as long as the structure of the environment does not change qualitatively, and as long as sensor and effector errors can be modeled in a tractable way. The SSH assumes that the environment

[6] The availability of accurate external positioning information such as GPS does not solve the mapping or localization problems, because of low resolution, limited signal accessibility, and unmapped or changing environments. GPS input is best regarded as a sensory feature useful for disambiguating certain places.

can be decomposed into overlapping regions, each of which is qualitatively uniform in the sense that an appropriately selected control law can operate effectively throughout it, coping with disturbances and detecting the qualitative change that signals the boundary of the region. In the most resource-limited case, to “operate effectively” means to be able to hill-climb to a distinctive state when in its neighborhood, or to be able to follow a trajectory from one distinctive state to the next. When resources are more plentiful, to “operate effectively” can include the ability to localize at arbitrary positions or to reach arbitrary positions within the current neighborhood.

The SSH works in environments where locally distinctive states can be defined, and where motion in qualitatively uniform regions is sufficiently reliable between locally distinctive states that the agent’s behaviors can be abstracted to the nodes and arcs of a graph. Local motion in the graph is then treated as perfectly reliable, insulated from the sensorimotor disturbances handled by the control laws.

Global uncertainty arises when collections of qualitatively uniform regions are non-simply connected. Traveling around a large loop typically accumulates substantial uncertainty about the metrical position of the agent with respect to the frame of reference of the starting point (see Fig. 7). The SSH decomposes this uncertainty into topological uncertainty (“Are we there yet?”) and metrical uncertainty (“How far are we, and in what direction?”).

Topological uncertainty is handled at the SSH causal and topological levels, as a subgraph-matching problem (Section 4.4). When the exploring agent arrives at a node of the graph (a place or distinctive state), it may need to determine whether this node is the same as a previously-encountered node. Some evidence may come from the sensory features available at the two nodes. Other evidence may come from comparing the topological neighbors of the two nodes. If the agent’s sensors can detect enough variety in the environment, this decision can be made to any desired degree of confidence, after which the global topological structure of the environment is treated as reliably known.

Once the topological structure of the set of reliable travel paths is reliably known, effective travel among distinctive states is possible even without accurate global metrical knowledge. Furthermore, global metrical uncertainty can be incrementally reduced with additional observations gathered during travel. Obviously, accurate global metrical knowledge is still useful, particularly for deducing and following new trajectories, and for coping with disasters such as getting blown off course by a storm at sea. But in the SSH, accurate global metrical knowledge is not on the critical path for most cases of mapping and navigation.

While the office environment is convenient for research, the SSH is intended for wider applicability, including woods, open fields, or the surface of Mars. Micronesian navigation [29,34] provides an extreme (and extremely interesting) example of this. Navigators in dugout canoes without modern instruments sail among the islands of the Micronesian archipelago, often out of sight of land for days or even weeks, orienting themselves with respect to another island (the *etak* island) that is *also* out of sight during the entire voyage! The methods they have for selecting safe routes, and for estimating heading, velocity and hence position, are fascinating. The critical point for this paper is that the navigators’ cognitive representation of the route is as a sequence of places linked by travel segments with different control laws. These places are described in terms of features that are not directly observable by the navigator, such as a supernatural event or the position of the *etak* island. The symbolic causal and topological representation of the route provides the representational framework that coordinates the metrical inferences that actually do the work.

The SSH representation is not universal. If the environment is sufficiently uniform, as perceived and represented by the agent, that there are no distinctive states or that travel among distinctive states is unreliable, then SSH mapping is impossible. Novices at sea, in the desert, in the woods, or even in Levittown suburban tracts [67], fail to build useful cognitive maps or to navigate with them.

A key claim about the SSH is that the different spatial knowledge representations and inference methods exist in order to cope with the different components of spatial uncertainty, and therefore to produce the kind of robust behavior seen in humans.

8.2. Recent successful robots

The development of the Spatial Semantic Hierarchy has taken place during a time of exciting progress in intelligent robot control, map-building and navigation. There has been a ferment of important ideas that are combined in different ways by different researchers. This progress is surveyed in an excellent collection of chapters, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, edited by Kortenkamp, Bonasso and Murphy [39]. The map-building systems surveyed are: RHINO, by Thrun and his colleagues at CMU and Bonn [93]; CARMEL, by Kortenkamp and his colleagues at Michigan [38]; DERVISH, by Nourbakhsh and colleagues at Stanford [76]; and XAVIER, by Koenig and Simmons at CMU [37]. The SSH and its precursors have both influenced, and been influenced by, these systems.

An occupancy grid with a single frame of reference is the primary representation for the spatial structure of the environment for RHINO, CARMEL, and XAVIER. DERVISH uses a purely topological map, using control laws to follow edges and recognize places much like the SSH. XAVIER uses a very large grid cell (1 m $\times$ 1 m), and explicitly links the sequences of cells along corridors, describing some of the topological structure of its environment. RHINO uses a Voronoi-diagram based analysis to infer a topological description of the environment from the completed metrical map, in order to get the more compact and efficient topological representation for problem-solving. From the SSH perspective, deriving the topological map from the global metrical map puts the most expensive and error-prone representation on the critical path before a much more robust, flexible, and inexpensive representation.

Localization for RHINO and CARMEL means to identify the coordinates of the robot within the global frame of reference. RHINO uses a Bayesian approach to identify the maximum likelihood location given sensor input and recent history. CARMEL determines and maintains its position within the global coordinate frame using triangulation from observations of landmarks at known locations. Both DERVISH and XAVIER use probabilistic state-set representations for the robot location, thereby tolerating any ambiguity in identifying the current place. DERVISH uses "assumptive planning", which derives plans assuming that the single most likely location is correct, but tracks their execution with the full state-set representation. XAVIER uses a uniform POMDP representation both to plan its actions and to assimilate observations. In the SSH, localization occurs at any or all of the four levels, representing the current control law; the current action or distinctive state; the current place, path and direction; and the current coordinates within the local frame of reference in the patchwork map. The rehearsal procedure for resolving topological ambiguity is effective but ad hoc, and should perhaps be replaced by a more principled POMDP-based strategy.

Local motion control and collision avoidance is done using a local occupancy grid in an egocentric frame of reference by CARMEL and XAVIER. RHINO uses a velocity-space representation of the immediate neighborhood that makes it possible to encode a variety of hard and soft constraints on motion. DERVISH uses a case-based representation that specifies how to move in environments matching known patterns. In all cases, the spatial representation for deriving local motion is distinct from the one used for mapping and localization in the environment as a whole.

Many intelligent mobile robot systems exploit a "three-layer architecture" [5,10,25], where the different layers implement processing methods running at different time-scales and drawing on different sources of information. The ontologies of the knowledge representations at the SSH levels corresponds well with the needs of the architectural layers. The SSH control level is well suited to the control layer of the architecture. The use of differential equations as the formalism for SSH control laws gives it access to the full power of traditional control theory. The SSH causal level corresponds to the sequencing layer, but the language of causal schemas has only a subset of the capabilities of programming languages such as RAPS [21] and ATLANTIS [26] used to implement the sequencing layer in robotic applications. The SSH topological and metrical levels provide declarative descriptions of the environment suitable for processing by modules in the deliberative layer.

There are many different layered and non-layered architectures that can be used to organize the processing stages in an intelligent mobile robot. The claim behind the SSH is that there are fundamental dependencies among the different types of spatial knowledge and their representation (Fig. 1), that must be respected by any architecture. It is this knowledge-level dependency that is responsible for the similarities among three-layer architectures.

8.3. Graphical maps and verbal route directions

The SSH, as presented in this paper, has focused on knowledge obtained as observations during travel through the environment. However, the SSH representation includes natural targets for information obtained from graphical maps or verbal directions [96].

Graphical maps correspond most straight-forwardly to the metrical level, where local metrical maps can align frames of reference with the graphical map, and allow conclusions to be mapped from the graphical medium back to the SSH metrical level. Topological routes can be found visually on a graphical map and translated to the SSH topological level. Causal level descriptions of routes can also be generated from simulated travel through a graphical map, but are notoriously subject to errors, particularly in the directions of turns.

Verbal route directions, whether spoken or typed, are frequently sequences of imperatives and their results, and so correspond naturally to knowledge at the SSH causal level.

Verbal descriptions of metrical relations are frequently inaccurate and unhelpful, but verbal descriptions of the situation for taking an action, or of the topological connections in the map are common and useful [66,85].

8.4. Localizability and reachability

Given an environment, an agent traveling within it, and the SSH as instantiated for its sensorimotor system, it should be possible to characterize each state s in the environment according to its accessibility in the cognitive map.

A state s is *localizable* if, starting from s, the agent can reliably reach any distinctive state in the topological map. This relies on s being in a region where a trajectory-following or hill-climbing control law can be selected, and being within the basin of attraction where that control law will lead to a distinctive state. It also requires the topological map to be connected, so access to one distinctive state provides reliable access to all the others. (Of course, these concepts could be defined to be relative to the current component of the map.)

A state s is *reachable* if, starting at any distinctive state in the environment, there is a reliable method for the agent to travel to s. If the topological map is connected, every distinctive state is reachable. If s lies on a path, and has a known position within the 1-D coordinate system of the path, it is reachable. Otherwise, if s has known coordinates within a 2-D local metrical map, it is reachable. Clearly, every reachable state is localizable, but not necessarily vice versa.

Using these concepts, one can envision analyzing the value of a particular sensorimotor system to exploration in a given environment, and analyzing how localizability and reachability change with the agent's experience in the environment as the cognitive map becomes more richly specified.

9. Conclusion

Like the famous story of the blind men examining the elephant, the cognitive map has appeared entirely differently to different observers. The Spatial Semantic Hierarchy proposes that this apparent heterogeneity is a real feature of the phenomenon, and indeed is the source of the flexibility, power and robustness of the cognitive map. The hierarchy of representations provides great expressive power for incomplete knowledge, provides targets for opportunistic (hence inexpensive) assimilation of new knowledge and inference from existing knowledge, and provides multiple knowledge sources for problem-solving under diverse circumstances.

The current paper provides an overview of the different representations and the implementations that evaluate their effectiveness. In future work, we plan to characterize the SSH more formally in logic (e.g., [83,84]), extend the SSH approach to encompass knowledge of visual space (e.g., [31,32]), and to explore the use of graphical maps and verbal route directions. Over the longer term, the ideas about the structure of knowledge embodied in the Spatial Semantic Hierarchy may be useful for the understanding of commonsense knowledge generally, and may be able to shed light on the neural substrate for spatial reasoning and on cross-species variation in spatial abilities.

References

[1] P.E. Agre, D. Chapman, Pengi: An implementation of a theory of activity, in: Proc. AAAI-87, Seattle, WA, 1987.

[2] R.C. Arkin, Motor schema-base mobile robot navigation, Internat. J. Robotics Res. 8 (4) (1989) 92–112.

[3] C.G. Atkeson, A.W. Moore, S. Schaal, Locally weighted learning, Artificial Intelligence Rev. 11 (1/5) (1997) 11–73.

[4] C.G. Atkeson, A.W. Moore, S. Schaal, Locally weighted learning for control, Artificial Intelligence Rev. 11 (1/5) (1997) 75–113.

[5] R.P. Bonasso, Integrating reaction plans and layered competences through synchronous control, in: Proc. IJCAI-91, Sydney, Australia, 1991, pp. 1225–1231.

[6] R.A. Brooks, Symbolic reasoning among 3D models and 2D images, Artificial Intelligence 17 (1981) 285–348.

[7] R.A. Brooks, Visual map making for a mobile robot, in: Proc. IEEE International Conference on Robotics and Automation, St. Louis, MO, 1985, pp. 824–829.

[8] R. Chatila, J.-P. Laumond, Position referencing and consistent world modeling for mobile robots, in: Proc. IEEE International Conference on Robotics and Automation, St. Louis, MO, 1985, pp. 138–170.

[9] E. Chown, S. Kaplan, D. Kortenkamp, Prototypes, location, and associative networks (PLAN): Towards a unified theory of cognitive mapping, Cognitive Sci. 19 (1) (1995) 1–51.

[10] J. Connell, SSS: A hybrid architecture applied to robot navigation, in: Proc. IEEE International Conference on Robotics and Automation, Nice, France, 1992.

[11] I.J. Cox, J.J. Leonard, Modeling a dynamic environment using a Bayesian multiple hypothesis approach, Artificial Intelligence 66 (1994) 311–344.

[12] J.M. Crawford, B.J. Kuipers, Negation and proof by contradiction in access-limited logic, in: Proc. AAAI-91, Anaheim, CA, 1991.

[13] T. Dean, D. Angluin, K. Basye, S. Engelson, L. Kaelbling, E. Kokkevis, O. Maron, Inferring finite automata with stochastic output functions and an application to map learning, in: Proc. AAAI-92, San Jose, CA, 1992.

[14] T. Dean, K. Basye, R. Chekaluk, S. Hyun, M. Lejter, M. Randazza, Coping with uncertainty in a control system for navigation and exploration, in: Proc. AAAI-90, Boston, MA, 1990.

[15] T.L. Dean, M.P. Wellman, Planning and Control, Morgan Kaufmann, San Mateo, CA, 1991.

[16] E.W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269–271.

[17] E.W. Dijkstra, A Discipline of Programming, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[18] R.M. Downs, D. Stea, Image and Environment, Aldine Publishing Company, Chicago, IL, 1973.

[19] G. Dudek, M. Jenkin, E. Milios, D. Wilkes, Robotic exploration as graph construction, IEEE Trans. Robotics and Automation 7 (6) (1991) 859–865.

[20] R.J. Elliot, M.E. Lesk, Route finding in street maps by computers and people, in: Proc. AAAI-82, Pittsburgh, PA, 1982.

[21] R.J. Firby, Task networks for controlling continuous processes, in: Proc. 2nd International Conference on AI Planning Systems, 1994.

[22] G.F. Franklin, J.D. Powell, A. Emami-Naeini, Feedback Control of Dynamic Systems, 3rd. edn., Addison-Wesley, Reading, MA, 1994.

[23] M.O. Franz, B. Schölkopf, H.A. Mallot, H.H. Bülthoff, Learning view graphs for robot navigation, Autonomous Robots 5 (1998) 111–125.

[24] C.R. Gallistel, The Organization of Learning, MIT Press, Cambridge, MA, 1990.

[25] E. Gat, Three-layer architectures, in: D. Kortenkamp, R.P. Bonasso, R. Murphy (Eds.), Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems, AAAI Press, Menlo Park, CA/MIT Press, Cambridge, MA, 1998, pp. 195–210.

[26] E. Gat, Integrating planning and reacting in a heterogenous asynchronous architecture for controlling real-world mobile robots, in: Proc. AAAI-92, San Jose, CA, 1992.

[27] A. Gelb, Applied Optimal Estimation, MIT Press, Cambridge, MA, 1974.

[28] S. Gillner, H.A. Mallot, Navigation and acquisition of spatial knowledge in a virtual maze, J. Cognitive Neuroscience 10 (4) (1998) 445–463.

[29] T. Gladwin, East is a Big Bird: Navigation and Logic on Puluwat Atoll, Harvard University Press, Cambridge, MA, 1970.

[30] J. Glasgow, N.H. Narayanan, B. Chandrasekaran, Diagrammatic Reasoning: Cognitive and Computational Perspectives, AAAI Press, Menlo Park, CA/MIT Press, Cambridge, MA, 1995.

[31] W.S. Gribble, Slow visual search in a fast-changing world, in: Proc. 1995 IEEE Symposium on Computer Vision (ISCV-95), 1995.

[32] W.S. Gribble, R.L. Browning, M. Hewett, E. Remolina, B.J. Kuipers, Integrating vision and spatial reasoning for assistive navigation, in: V. Mittal, H. Yanco, J. Aronis, R. Simpson (Eds.), Assistive Technology and Artificial Intelligence, Lecture Notes in Computer Science, Springer, New York, 1998.

[33] S. Harnad, The symbol grounding problem, Physica D 42 (1990) 335–346.

[34] E.L. Hutchins, Cognition in the Wild, MIT Press, Cambridge, MA, 1995.

[35] S.S. Iyengar, A. Elfes (Eds.), Autonomous Mobile Robots: Perception, Mapping, and Navigation, IEEE Computer Society Press, Los Alamitos, CA, 1991.

[36] T. Jochem, D. Pomerleau, Life in the fast lane: The evolution of an adaptive vehicle control system, AI Magazine 17 (2) (1996) 11–50.

[37] S. Koenig, R.G. Simmons, Xavier: A robot navigation architecture based on partially observable Markov decision process models, in: D. Kortenkamp, R.P. Bonasso, R. Murphy (Eds.), Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems, AAAI Press, Menlo Park, CA/MIT Press, Cambridge, MA, 1998, pp. 91–122.

[38] D. Kortenkamp, M. Huber, C. Cohen, U. Raschke, F. Koss, C. Congdon, Integrating high-speed obstacle avoidance, global path planning and vision sensing on a mobile robot, in: D. Kortenkamp, R.P. Bonasso, R. Murphy (Eds.), Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems, AAAI Press, Menlo Park, CA/MIT Press, Cambridge, MA, 1998, pp. 53–71.

[39] D. Kortenkamp, R.P. Bonasso, R. Murphy (Eds.), Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems, AAAI Press, Menlo Park, CA/MIT Press, Cambridge, MA, 1998.

[40] A. Kosaka, A.C. Kak, Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties, Computer Vision, Graphics, and Image Processing-Image Understanding 56 (1992) 271–329.

[41] B. Kosko, Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[42] S.M. Kosslyn, Image and Mind, Harvard University Press, Cambridge, MA, 1980.

[43] S.M. Kosslyn, Image and Brain, MIT Press, Cambridge, MA, 1994.

[44] S.M. Kosslyn, S.P. Shwartz, A simulation of visual imagery, Cognitive Sci. 1 (1977) 265–295.

[45] B.J. Kuipers, Representing knowledge of large-scale space, Ph.D. Thesis, Massachusetts Institute of Technology, Mathematics Department, Cambridge, MA, 1977; also MIT-AI TR-418, 1977. Available via ftp://ftp.cs.utexas.edu/pub/qsim/papers/Kuipers-PhD-77.ps.gz.

[46] B.J. Kuipers, Modeling spatial knowledge, Cognitive Sci. 2 (1978) 129–153; reprinted in: Su-Shing Chen (Ed.), Advances in Spatial Reasoning, Vol. 2, Ablex Publishing, Norwood, NJ, 1990, pp. 171–198.

[47] B.J. Kuipers, Commonsense knowledge of space: Learning from experience, in: Proc. IJCAI-79, Tokyo, Japan, 1979, pp. 499–501; reprinted in: Su-Shing Chen (Ed.), Advances in Spatial Reasoning, Vol. 2, Ablex Publishing, Norwood, NJ, 1990, pp. 199–206.

[48] B.J. Kuipers, The 'Map in the Head' metaphor, Environment and Behavior 14 (1982) 202–220.

[49] B.J. Kuipers, The cognitive map: Could it have been any other way?, in: H.L. Pick Jr., L.P. Acredolo (Eds.), Spatial Orientation: Theory, Research, and Application, Plenum Press, New York, 1983.

[50] B.J. Kuipers, Modeling human knowledge of routes: Partial knowledge and individual variation, in: Proc. AAAI-83, Washington, DC, 1983.

[51] B.J. Kuipers, Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge, MIT Press, Cambridge, MA, 1994.

[52] B.J. Kuipers, A hierarchy of qualitative representations for space, in: Proc. 10th International Workshop on Qualitative Reasoning about Physical Systems, Fallen Leaf Lake, CA, AAAI Technical Report WS-96-01, AAAI Press, Menlo, Park, CA, 1996, pp. 113–120.

[53] B.J. Kuipers, A hierarchy of qualitative representations for space, in: C. Freksa, C. Habel, K.F. Wender (Eds.), Spatial Cognition: An Interdisciplinary Approach to Representing and Processing Spatial Knowledge, Lecture Notes in Artificial Intelligence, Vol. 1404, Springer, Berlin, 1998, pp. 337–350.

[54] B.J. Kuipers, K. Åström, The composition and validation of heterogeneous control laws, Automatica 30 (2) (1994) 233–249.
[55] B.J. Kuipers, Y.T. Byun, A robust qualitative method for spatial learning in unknown environments, in: Proc. AAAI-88, St. Paul, MN, Morgan Kaufmann, Los Altos, CA, 1988, pp. 774–779.
[56] B.J. Kuipers, Y.-T. Byun, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations, Robotics and Autonomous Systems 8 (1991) 47–63.
[57] B.J. Kuipers, R. Froom, W.-Y. Lee, D. Pierce, The semantic hierarchy in robot learning, in: J. Connell, S. Mahadevan (Eds.), Robot Learning, Kluwer Academic, Dordrecht, 1993, pp. 141–170.
[58] B.J. Kuipers, T. Levitt, Navigation and mapping in large scale space, AI Magazine 9 (2) (1988) 25–43; reprinted in: Su-Shing Chen (Ed.), Advances in Spatial Reasoning, Vol. 2, Ablex Publishing, Norwood, NJ, 1990, pp. 207–251.
[59] G. Lakoff, M. Johnson, Metaphors We Live By, The University of Chicago Press, Chicago, IL, 1980.
[60] J.-C. Latombe, Robot Motion Planning, Kluwer Academic, Boston, MA, 1991.
[61] X. Lebègue, J.K. Aggarwal, Generation of architectural CAD models using a mobile robot, in: Proc. IEEE International Conference on Robotics and Automation, San Diego, CA, 1994.
[62] W.Y. Lee, Spatial semantic hierarchy for a physical mobile robot, Ph.D. Thesis, University of Texas at Austin, Computer Science Department, 1996; Available via http://www.cs.utexas.edu/users/qr/robotics/papers.html.
[63] J.J. Leonard, H.F. Durrant-Whyte, Directed Sonar Sensing for Mobile Robot Navigation, Kluwer Academic, Boston, MA, 1992.
[64] T.S. Levitt, D.T. Lawton, Qualitative navigation for mobile robots, Artificial Intelligence 44 (3) (1990) 305–360.
[65] V. Lifschitz, Nested abnormality theories, Artificial Intelligence 74 (1995) 351–365.
[66] K.L. Lovelace, M. Hegarty, D.R. Montello, Elements of good route directions in familiar and unfamiliar environments, in: C. Freksa, D.M. Mark (Eds.), Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science, Lecture Notes in Computer Science, Vol. 1661, Springer, Berlin, 1999, pp. 65–82.
[67] K. Lynch, The Image of the City, MIT Press, Cambridge, MA, 1960.
[68] E.H. Mamdani, Applications of fuzzy algorithms for control of a simple dynamic plant, Proc. IEE 121 (1974) 1585–1588.
[69] M.J. Mataric, Integration of representation into goal-driven behavior-based robots, IEEE Trans. on Robotics and Automation 8 (3) (1992) 304–312.
[70] J. McCarthy, P.J. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence 4, Edinburgh University Press, Edinburgh, 1969, pp. 463–502.
[71] D.V. McDermott, E. Davis, Planning routes through uncertain territory, Artificial Intelligence 22 (1984) 107–156.
[72] H. Moravec, A. Elfes, High resolution maps from wide angle sonar, in: Proc. IEEE International Conference on Robotics and Automation, St. Louis, MO, 1985, pp. 116–121.
[73] H.P. Moravec, Sensor fusion in certainty grids for mobile robots, AI Magazine 9 (1988) 61–74.
[74] R. Nevatia, T.O. Binford, Description and recognition of curved objects, Artificial Intelligence 8 (1) (1977) 77–98.
[75] N.J. Nilsson, Principles of Artificial Intelligence, Tioga Publishing, Palo Alto, CA, 1980.
[76] I. Nourbakhsh, Dervish: An office-navigating robot, in: D. Kortenkamp, R.P. Bonasso, R. Murphy (Eds.), Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems, AAAI Press, Menlo Park, CA/MIT Press, Cambridge, MA, 1998, pp. 73–90.
[77] J. O'Keefe, L. Nadel, The Hippocampus as a Cognitive Map, Oxford University Press, New York, 1978.
[78] J.S. Penberthy, D.S. Weld, Temporal planning with continuous change, in: Proc. AAAI-94, Seattle, WA, 1994, pp. 1010–1015.
[79] J. Piaget, B. Inhelder, The Child's Conception of Space, Norton, New York, 1967; First published in French, 1948.
[80] D. Pierce, B. Kuipers, Learning to explore and build maps, in: Proc. AAAI-94, Seattle, WA, 1994.
[81] D.M. Pierce, B.J. Kuipers, Map learning with uninterpreted sensors and effectors, Artificial Intelligence 92 (1997) 169–227.

[82] D.A. Pomerleau, Knowledge-based training of artificial neural networks for autonomous robot driving, in: J.H. Connell, S. Mahadevan (Eds.), Robot Learning, Kluwer Academic, Dordrecht, 1993.
[83] E. Remolina, B. Kuipers, Boundary region relations, in: Cognitive Robotics Papers from the 1998 AAAI Fall Symposium, AAAI Technical Report FS-98-02, AAAI Press, Menlo Park, CA, 1998, pp. 117–124.
[84] E. Remolina, B. Kuipers, Towards a formalization of the spatial semantic hierarchy, in: Proc. 4th Symposium on Logical Formalizations of Commonsense Reasoning (Common Sense 98), London, England, 1998, pp. 126–137.
[85] C.K. Riesbeck, "You can't miss it!": Judging the clarity of directions, Cognitive Sci. 4 (1980) 285–303.
[86] B. Schölkopf, H.A. Mallot, View-based cognitive mapping and path planning, Adaptive Behavior 3 (1995) 311–348.
[87] M.J. Schoppers, Universal plans for reactive robots in unpredictable environments, in: Proc. IJCAI-87, Milan, Italy, Morgan Kaufmann, San Mateo, CA, 1987.
[88] R.N. Shepard, J. Metzler, Mental rotation of three-dimensional objects, Science 171 (1971) 701–703.
[89] A.W. Siegel, S.H. White, The development of spatial representations of large-scale environments, in: H.W. Reese (Ed.), Advances in Child Development and Behavior, Vol. 10, Academic Press, New York, 1975.
[90] M.G. Slack, Navigation templates: Mediating qualitative guidance and quantitative control in mobile robots, IEEE Trans. Systems Man Cybernet. 23 (2) (1993) 452–466.
[91] R. Smith, P. Cheeseman, On the representation of and estimation of spatial uncertainty, Internat. J. Robotics Research 5 (1986) 56–68.
[92] S.S. Stevens, On the theory of scales of measurement, Science 103 (1946) 677–680.
[93] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, T. Schmidt, Map-learning and high-speed navigation in RHINO, in: D. Kortenkamp, R.P. Bonasso, R. Murphy (Eds.), Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems, AAAI Press, Menlo Park, CA/MIT Press, Cambridge, MA, 1998, pp. 21–52.
[94] S. Thrun, D. Fox, W. Burgard, A probabilistic approach to concurrent mapping and localization for mobile robots, Machine Learning 31 (1–3) (1998) 29–53.
[95] S. Thrun, S. Gutmann, D. Fox, W. Burgard, B.J. Kuipers, Integrating topological and metric maps for mobile robot navigation: A statistical approach, in: Proc. AAAI-98, Madison, WI, 1998, pp. 989–995.
[96] B. Tversky, P.U. Lee, Pictorial and verbal tools for conveying routes, in: C. Freksa, D.M. Mark (Eds.), Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science, Lecture Notes in Computer Science, Vol. 1661, Springer, Berlin, 1999, pp. 51–64.
[97] S. Ullman, High-Level Vision: Object Recognition and Visual Cognition, MIT Press, Cambridge, MA, 1996.
[98] P. van Turennout, G. Honderd, L.J. van Schelven, Wall-following control of a mobile robot, in: Proc. IEEE International Conference on Robotics and Automation, Nice, France, 1992, pp. 280–285.
[99] R. Wehner, Large-scale navigation: The insect case, in: C. Freksa, D.M. Mark (Eds.), Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science, Lecture Notes in Computer Science, Vol. 1661, Springer, Berlin, 1999, pp. 1–20.